Proceedings of the
6th annual Advance Technology Workshop

ATW'98

19-20TH OF MAY, 1998

Hosted By
University of Corsica

Sponsored By
University of Corsica
Collectivité Territoriale de Corse
Consortium MicroLabs International
United States Air Force

We wish to thank the following for their contribution to the success of this conference :

United States Air Force
European Office of Aerospace Research and Development

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	22 June 1998	Conference Proceedings

4. TITLE AND SUBTITLE

Sixth Annual Advanced Technology Workshop ATW'98 Europe

5. FUNDING NUMBERS

F61775-98-WE037

6. AUTHOR(S)

Conference Committee

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Universite de Corse
URA CNRS 2053
Quartier Grossetti
CORTE 20250
France

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
PSC 802 BOX 14
FPO 09499-0200

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

CSP 98-1060

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

A

13. ABSTRACT (Maximum 200 words)

The Final Proceedings for Sixth Annual Advanced Technology Workshop ATW'98 Europe, 19 May 1998 - 20 May 1998

This is an interdisciplinary conference. Topics include systems design, application and tech transfer with the main focus on enhancing technology utilization. Specific areas include embedded systems, commercial standards, modeling/simulation, space applications, and industrial networks.

14. SUBJECT TERMS

Test & Evaluation, Active Control, Modelling & Simulation

15. NUMBER OF PAGES

165

16. PRICE CODE

N/A

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19, SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Main Topic : Enhancing Technology Utilization

For six years ATW has been the recognized source of applied collaboration between academia and industry. Government and industry participants have realized measurable improvements in productivity and quality, through a better prepared workforce, with the academic knowledge and practical training required to be competitive now.

Research topics presented by professors and graduate students give industry insight into developing technologies. While technology application topics, presented by industry, provide academia and government milestones by which to measure their effectiveness. And policy discussions provide academia and industry a window into future government support.

This year's topics will again focus on the different levels of technology application from research, to productization, and finally to commercialization. ATW will challenge you with the questions of tomorrow, while applying technology to your issues of today. Through the promotion of better academic planning by in-depth understanding of current industrial trends and future technology demands, ATW is the workshop where the future is designed.

ATW'98, Program Chair

J-F. SANTUCCI
URA CNRS 2053, Université de Corse
Quartier Grossetti, BP 52, 20250 CORTE, FRANCE
phone: (33) 4 95 45 01 26 / fax: (33) 4 95 61 05 51
e-mail: santucci@univ-corse.fr

ATW'98 Steering Committee

W. Ellis, Software Process and Metrics, USA
A. Rucinski, University of New Hampshire, USA
J-F. Santucci, University of Corse, France,
R.L. Straitt, United Defense, USA

ATW'98 Program Committee:

P. Arato, Techn. Univ. Budapest, Hungary
J.H. Balbi, President, University of Corsica, France
P. Bisgambiglia, University of Corsica, France
J. Costa, University of Corsica, France
D. Erbschloe, US Air Force, USA
M. Filippi , INRA Toulouse, France
D. Forrest, Univ. New Hampshire, EOS, USA
W. Jaworski, Concordia University, Canada
H. Joiner, SRC, Inc., USA
D. Hummer, Univ. of Duisburg, Germany
K. Olcoz, Univ. Complutense of Madrid, Spain
M. Oussalah, EMA-EERIE, France
B. Rucinska, Univ. New Hampshire, USA
S. Tewksbury, University West Virginia, USA

Contents

Technical Papers & slides of the corresponding presentation

University Courses Meet Software Engineering Standards

L. Andres, V. Jovanovic, B. Sherlund

University of Detroit Mercy

Abstract

This paper presents a first draft of a model for the transfer of best practices as represented by software engineering standards into university courses. Discussion of mechanisms for their use in the educational process (and the selection of standards actually used at the University of Detroit Mercy graduate program in Software Management) are presented as an example of the possibilities for the implementation of a model. Current work on assessing the use of software standards in education and preparation for the survey of software related programs as it related to the issues raised by the paper are also illustrated.

Introduction

Our work on the uses of software engineering standards as a mean to assure technology transfer from best industry practices as captured by software engineering and closely related standards into university programs is guided by the following premises:

1. Standards capture a common body of knowledge in Software Engineering as the state of the best industrial practice.

2. Software standards reached the state of being integrated into a systematic set, with consistent terminology and with a reasonably complete coverage of the software work.

3. Students working on projects (in teams) use standards to learn about the relevant aspects of software work relating previous experiences and the lectures.

4. Students use standards directly for defining work products, plans etc. in order to produce a software products deliverables (and interim products) working on the projects.

Model

Our model aim is to relate expected depth of knowledge with key ideas, references, and during the delivery, and in the actual transfer, use of (standardized) examples loaded with didactics.

In a curriculum oriented toward software development (Software Engineering, Computer Information Systems or similar) core competency is systematic software design, this carries both process and product focus. Courses should offer a balance of design principles, their applications and actual experiences with products (existing, reusable artifacts and to be made artifacts as well). They should also expose learners to contemporary methods and tools, and common industry practices as captured by standards as much as possible. Difficulties in designing standard courses are compounded by a range of didactical considerations and available educational technologies. The teaching model proposed here was influenced by the need for rigor and elegance, the knowledge competency levels to be attained and didactical and technological constraints. The intent is to draw other educators into the debate, thereby reviewing the multi-faceted aspects influencing design of software engineering courses. The model proposed in this paper is based on over 40 years of combined experience (of the authors following various didactical approaches) in teaching software engineering related courses. A number of delivery modes have been used in these courses, and coverage of software design and implementation had varied in definition and scope. In order to relate a course structure and content to the attainment of a predetermined depth of understanding of design, we adapted the cognitive knowledge levels of Bloom's Taxonomy (from six) to five levels of understanding. The knowledge level notion serves as a powerful metaphor to convey the intent of a course, and includes generic design competencies. Our choice of five levels has an analogy with capability maturity model (CMM) (1,3) and personal competency model (2). Whereas CMM levels provide a frame of reference for organizations and the personal software process or PSP (1) systematically disciplines individuals, the focus of our model is technical competence in software design.

Knowledge Levels

Understanding the work involved when designing a software system by applying the object-oriented paradigm involves a grasp of certain key concepts as well as engineering principles. This includes the software engineering terminology, modeling notations and an understanding of the underlying paradigms in terms of modeling concepts, primitives and relationships among them. This understanding allows one to analyze the techniques and methods, and apply them within the phases of selected software development life cycle (SDLC). The methods are supported by computer-aided environments, offering tools for performing technical tasks and for managing the project. A study of design patterns (topologies) and guidelines on how to apply these to new situations would complement the arsenal of knowledge in the form of concrete competencies. A number of standards representing a common body of knowledge has emerged. Understanding the UML standard (4,5) in particular, and building experience in applying it in controlled class room situations goes a long way toward preparing the budding professional.

The proposed knowledge levels are summarized as:

- Level-1: Software engineering terminology
- Level-2: Understanding modeling paradigms, notations, and relationships
- Level-3: Ability to analyze and model problems and solutions using appropriate methods
- Level-4: Build a record of quantitatively verified designs and implementations
- Level-5: Optimize software (design) technology through innovative skills.

Illustrative Implementation

An actual situation, at the University of Detroit Mercy graduate program in Software Management can be used as a fair illustration of our proposed model. As our primary interest and the focus of this paper is transfer of best practices (standards) into university courses, attention is to be shifted to the level of the entire curricula. For all the practical purposes a model set of standards encompassing software work is the IEEE set of Software Engineering Standards, an alternative could be a ESA set but that one can also be traced clearly to the same source. To say that the model is derived only from the experiences gained in incorporating, practices and corresponding standards (in defining and refining aforementioned program) will be an exaggeration but the fact is that a driving force in improving, standardizing the curricula was the effort to align the teaching with state of the better practices i.e. standards.. The table (below) provides full coverage of our program and illustrates the use of all the major standards.

Table-1: Sample of courses with a list of used standards

Courses	Standards and Guidelines
598 International Software Standards	ISO 15504, Trillium, ISO 9001
535 Software Metrics	IEEE 1061, ISO 9126, IEEE 982.1/2, IEEE 1045,
555 Databases	IDEF1X, SQL-92, ODMG-93
530 Software QA and Testing	ISO 9000-3, IEEE 829, IEEE 1008, IEEE 1044, IEEE 1044.1, TickIT, IEEE 1298, IEEE 1042,
540 Software Management	ISO 12207, DOD 2167A, MIL 498, IEEE 1074, CMM, IEEE 1012, IEEE 1228, IEEE 828, IEEE 1042, NASA.
556 Systems Analysis and Design	UML, IDEF0, IDEF1X
554 Software Maintenance	IEEE 1219,
520 Software Requirements Specification	IEEE 830, IDEF0, UML,
525 Software Design and Construction	UML, IEEE 1016, IEEE 1016.1,
510 Object Oriented Programming	UML, ANSI C++ STL,
505 Software Project Management	PMI BOK, IEEE 1058, BSI PM Guideline, PRINCE, IEEE 1062,

All the courses inherit the generic structure as follows:

- Model Course preamble (Purpose of course, Goal in terms of knowledge level attainment, Objectives in terms of expected outcomes, and Course focus)
- Model Syllabus, listing five educational units: 1. Prerequisite, 2. Basic, 3. Core, 4. Advanced and 5. Integration.

These units do not have to be offered in sequence, since the student's academic background and level of experience would be a determinant. The knowledge and skills attained would be assessed continually during the offering of Units 2, 3 and 4, whereas pre-assessment will be done at the start of Unit I and post-assessment after completing Unit 5.

- Model Unit-content, where each unit should address the following: (Purpose, Key references, Key concepts, issues and ideas, and Examples).
- Course didactics specify the role and responsibilities of the educator during the course, the role and responsibilities of students, purpose and nature of the individual assignments and team assignments; it also explains the role of the reference resources, the hardware and software systems and design tools available for the course.
- Delivery mode- specifies the teaching and learning modes of educators and students, the role of educational technology in the acquisition of knowledge and skills.

Survey

The purpose of this survey, sponsored by the IEEE Computer Society, is to collect the data about the use of Software Engineering Standards in regular University Courses, in order to provide support for the use of professional Software Engineering Standards in software education.

This survey focus is on the granularity (i.e., course level, as the key "experimental unit"), at which the use of software related standards will be investigated. A collection of courses makes a program curricula. A program curricular needs in standards can be systematized (addressed) and used as a consideration for accreditation and as (part of a minimal education) prerequisite for professional recognition of software engineers.

The primary goals of the survey being conducted is to collect empirical data about the use of software related standards in the formal education system. Another primary goal is to raise awareness of the possibilities by listing available standards per course(prompting).

The secondary goals are: 1. To solicit opinions about the use of software related standards in promoting the software engineering profession; 2. To investigate recognition of programs and software engineers as related to the use of the Software Engineering Standards; 3. To collect the data on textbooks now in wide use for software engineering courses.

This survey will be distributed to Colleges and Universities offering Software Engineering, Computer Science, Computer Information Systems, System Engineering, Computer Engineering and other related programs. Any program offering software related courses qualifies if it might benefit from the use of Software Engineering Standards.

The deliverables of this survey include the questionnaire and an invitation letter. The questionnaire will be sent out as an electronic mailing. The data will be collected by a IEEE Computer Society Web Page and linked to a repository at the University of Detroit Mercy, where the data will be available for analysis. The data will be collected in a database and the statistics will be shared with respondents. The database used will be Microsoft Access and Microsoft Excel spreadsheets will show the statistics.

Conclusions

The education sector prepares students for work related to software development. This encompasses university programs having courses related to software engineering. Introducing software engineering standards into course university curricula serves two purposes: (1) Harmonizing the body of knowledge; (2) Preparing both educational programs and professionals for certification. Our thesis is that professional standards codify the common body of knowledge (best practices). Teachers of software engineering and related university programs that prepare software professionals should become aware of software standards and guidelines. With their use in education, a new direction will be taken toward better quality and safer software. Exposing students to standards in their formative years will go a long way toward establishing a software engineering discipline and a true profession.

References

1. W. Humphrey "Managing the Software Process" Addison Wesley 1989,
2. W. Humphrey "A Discipline of Software Engineering" Addison Wesley 1995,
3. M. Paulk, ... "Capability Maturity Model for Software V1.1" CMU/SEI-93-TR-24,
4. G. Ford "Undergraduate programs in Software Engineering" SEI/CMU ...
5. V. Jovanovic, D. Olateju "Software Engineering- is there a Common Core" IBSCA Conference 1997,
6. V. Jovanovic, D. Shoemaker "Building Better Software Organization" Fresh Communication 1998,
7. V. Jovanovic "Software Engineering Standards and Education", position paper, SES Denver, 1994,

Measuring for Success!

"An effective way of implementing process improvements."

Presented at

Advanced Technology Workshop '98

University of Corsica - Ajaccio, Corsica (France)

19-20 May 1998

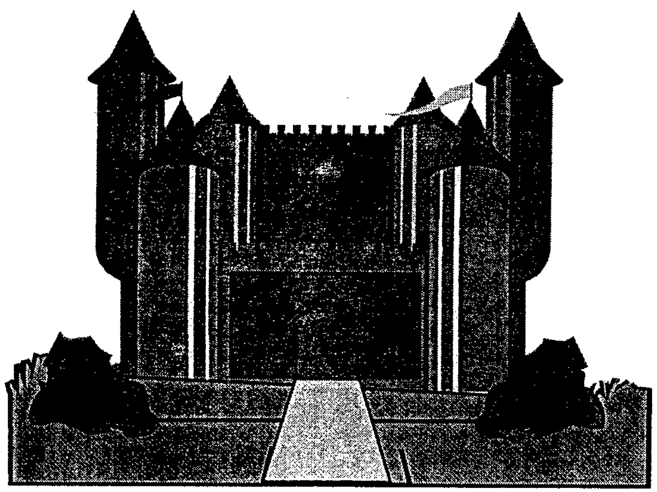
by

Robert L. Straitt

Measuring for Success

"An effective way of implementing process improvements."

Measurement, it is probably fair to say, is the cornerstone of knowledge. It allows us to compare things with other things to quantify relationships.*



Mine is bigger then yours amounts to a mathematical statement (M>Y)*

*K.C. Cole,1998

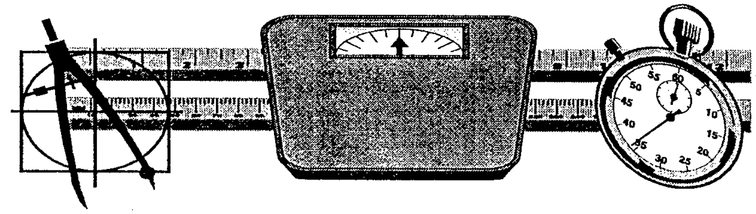
Measuring for Success

Discussion Topics

- Nature of Measurements
- Structure of a Measurement program
- Fundamentals of Process Improvement
- Integrating Measurements with Processes
- Pitfalls to Avoid
- Conclusion

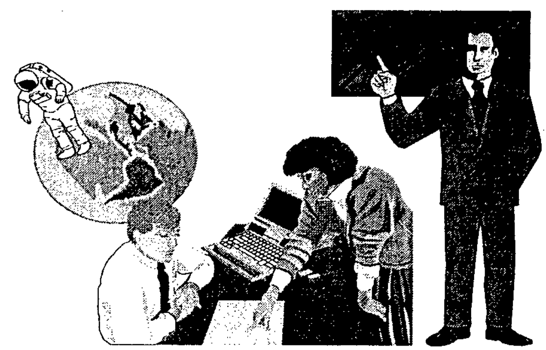
Nature of Measurements

What is measurement?



What is the role of measurement?

Who is measurement for?



Nature of Measurements

What is Measurement?

When you can measure…and express it in numbers, you know something about it. But, when you cannot…, your knowledge is of a meager and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely…advanced to the stage of science.

Lord Kelvin

Nature of Measurements

What is Measurement?

In the contest of engineering a measure is: *A quantitative assessment of the degree to which a product or process processes a given attribute.*

IEEE Standard 982.1 1988

"Measurement is a well established practice that transcends all engineering disciplines.... Measurement enables an organization to learn, to navigate, and to manage its rate of change."

Lloyd Moseman, Deputy Assistant Secretary of the Air Force, 1995

Nature of Measurements

What is Measurement?

- In the context of business measurement is: *A careful evaluation and documenting of the profitability of an organization's business processes.* *Robert L. Straitt, 1998*

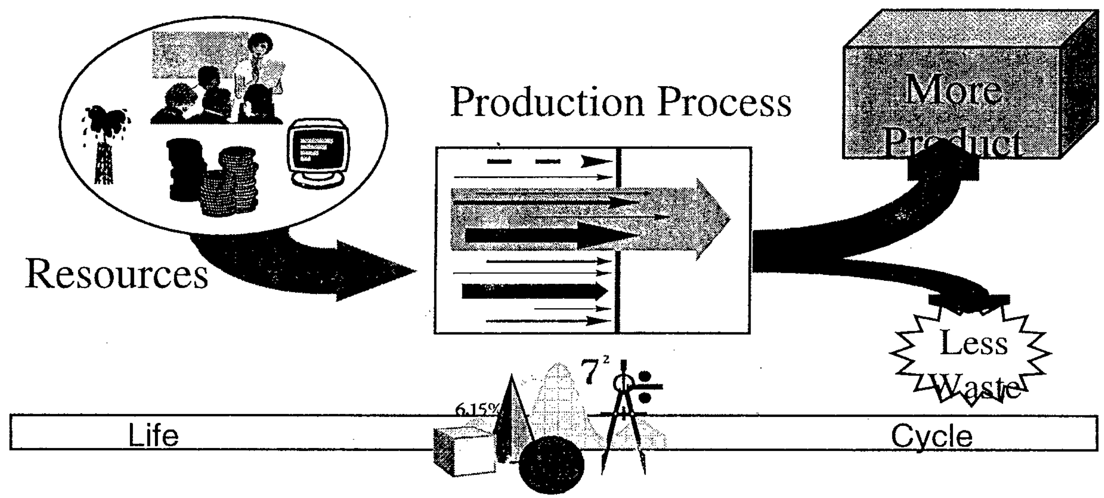
- The demand for measurement, generally in terms of money is inherent to every enterprise… Top management usually requires justification and rejustification…. it is necessary to be convincing not only in terms of logic, but in real money. *The Economics of Standardization, 1984*

Nature of Measurements

What is the Role of Measurement?

Identify Variation

"The central problem of management in all its aspects, including planning, procurement, manufacturing, research, sales, personnel, accounting, and law , is to understand better the meaning of variation, and to extract the information contained in variation" **Lloyd S. Nelson**

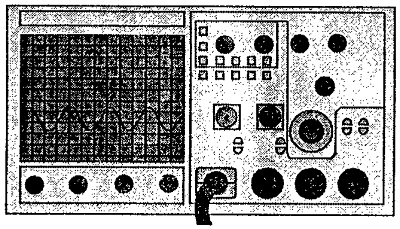


Measurement Process

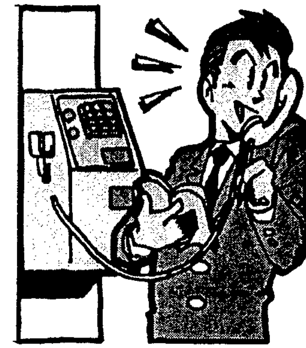
Nature of Measurements

What is the Role of Measurement?

Mechanisms of Measurement

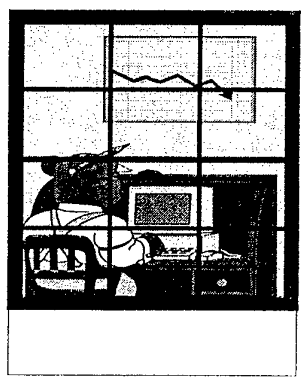


Understanding of
the current state

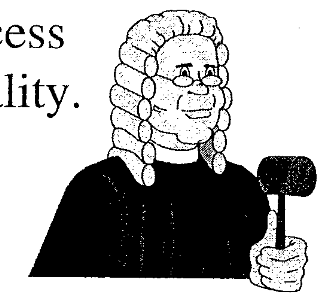


Communication
mechanism
to feed back
progress

Window into business
processes for
decision
makers



Independent verification
of business process
validity and vitality.

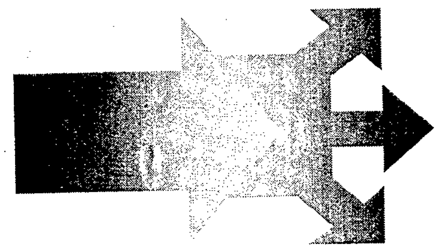


Nature of Measurements

What is the Role of Measurement?

Business Process Improvement

- Early problem identification and elimination.
- Accurate estimating, planning and implementation.
- Clear insight into business activities and processes.



- Customer satisfaction.
- Customer confidence.
- Less rework.
- More business growth.
- Higher Profits.

"Without the right information, you're just another person with an opinion." Tracy O'Rourke, CEO of Allen-Bradley

Nature of Measurements

What is the Role of Measurement?

Measurement Means Greater Returns

Create a Management Baseline

Provide a Window Into Organization for Senior Management

Track Progress Towards Business Goals

Monitor Return on Investments

Recommend Processes for Improvement Efforts

Determine Effectiveness of Improvement Efforts

Identify Obsolete Processes

Isolate Processes With No Throughput

THE BOTTOM LINE

Lower Costs, Better Quality, Higher Productivity

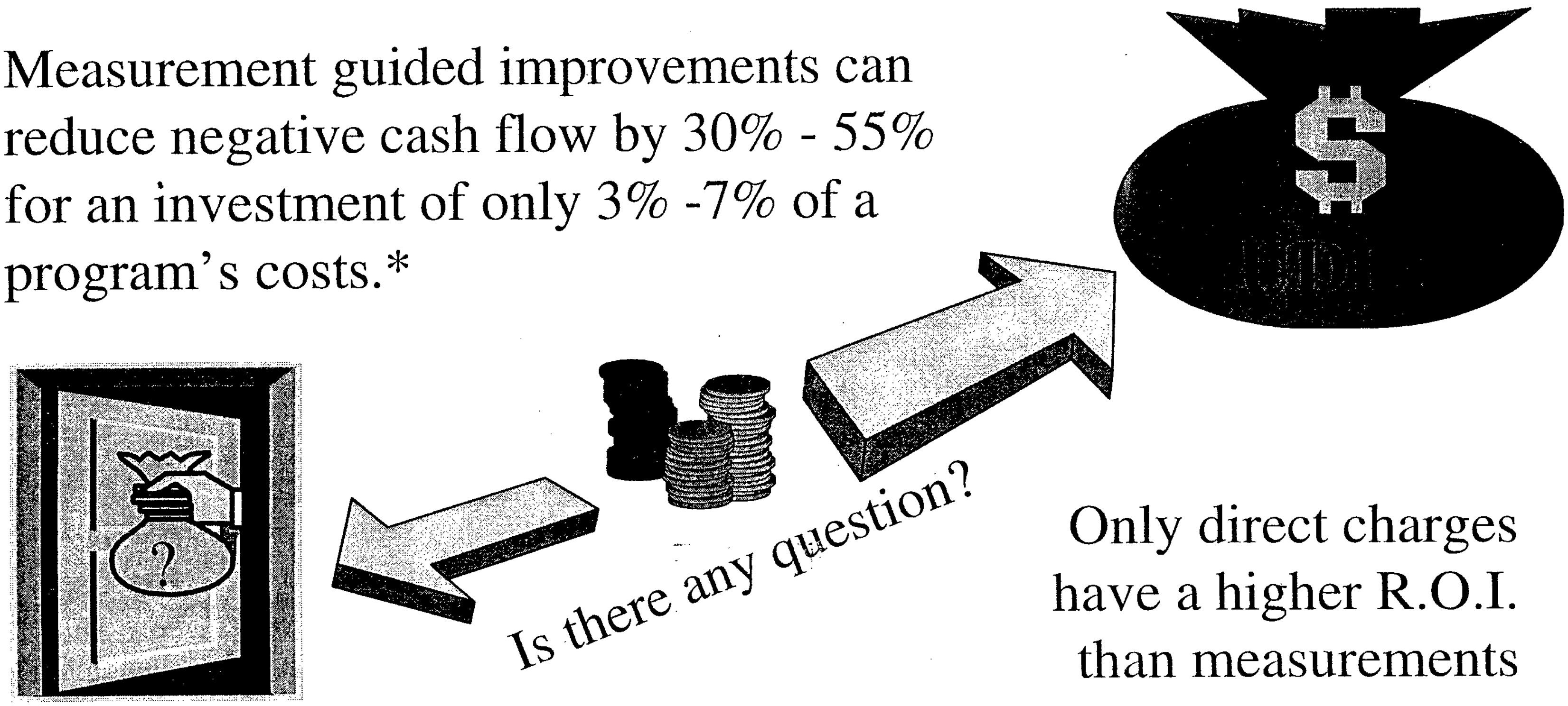
MORE PROFIT

Nature of Measurements

What is the Role of Measurement?

Measurements and Cash Flow

Measurement guided improvements can reduce negative cash flow by 30% - 55% for an investment of only 3% -7% of a program's costs.*

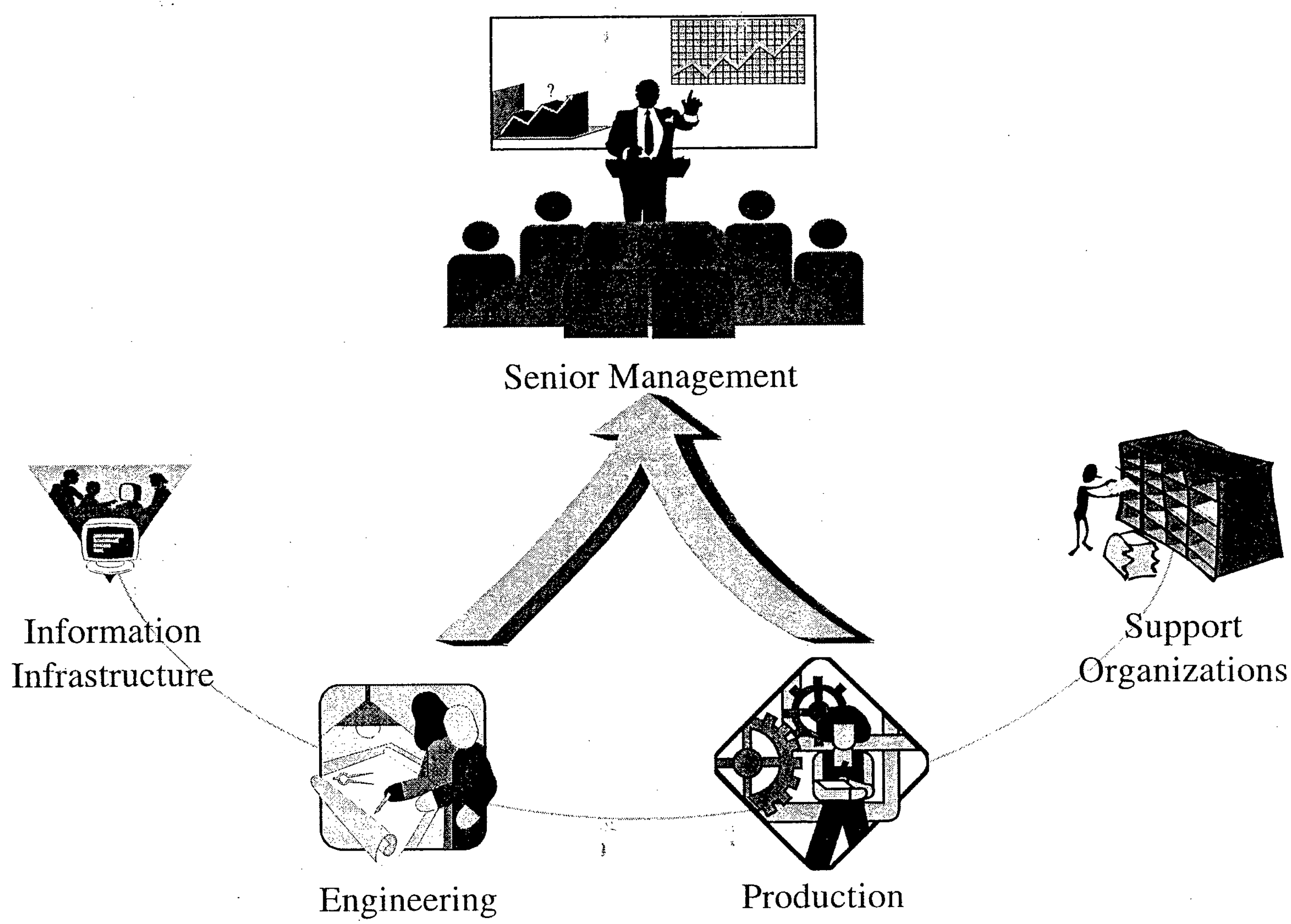


Only direct charges have a higher R.O.I. than measurements

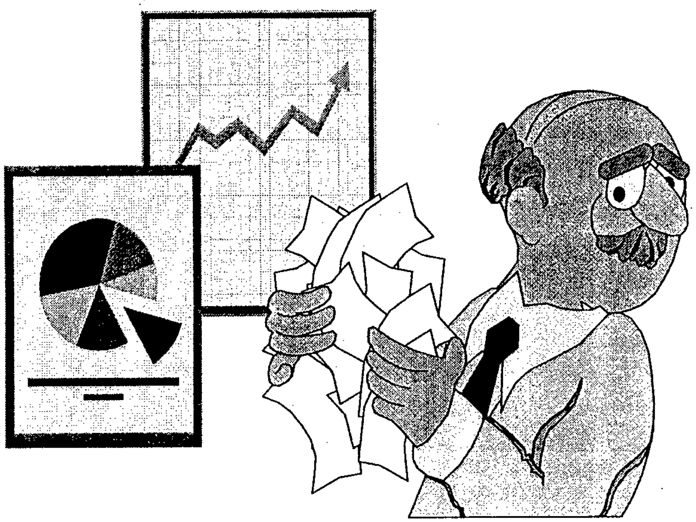
*NASA, USA and USAF data showing cost and savings of applying management measures.

Nature of Measurements

Who Measurements support.!



Structure of a Measurement Program



- Information Hierarchy

  The level of abstraction of information.

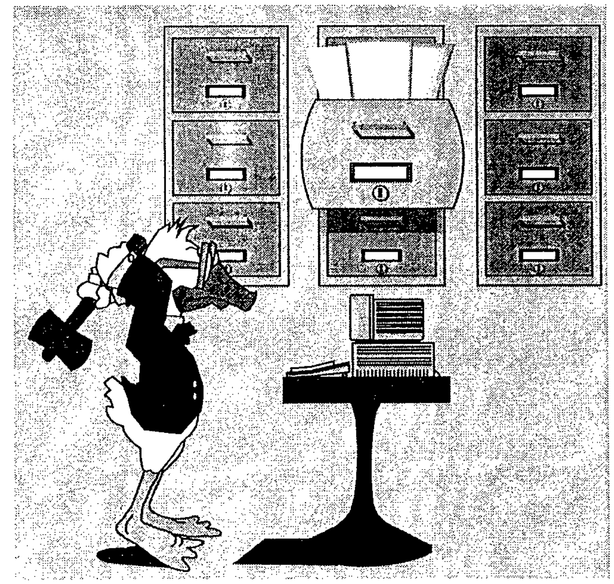
- Program Based Approach

Structure of a Measurement Program

The Information Hierarchy.

Information overload is the most serious handicaps facing today's business manager and government executive... The proper use of information is the hallmark of a successful manager.

Morton F. Meltzer, 1981



Have things improved in the last twenty years?

"In spite of the abundance of information, or maybe partly because of it, the West has great difficulty in finding its bearings amid contemporary events."

Aleksander I. Solzhenitsyn, Commencement, Harvard University 1978

Structure of a Measurement Program

The Information Hierarchy.

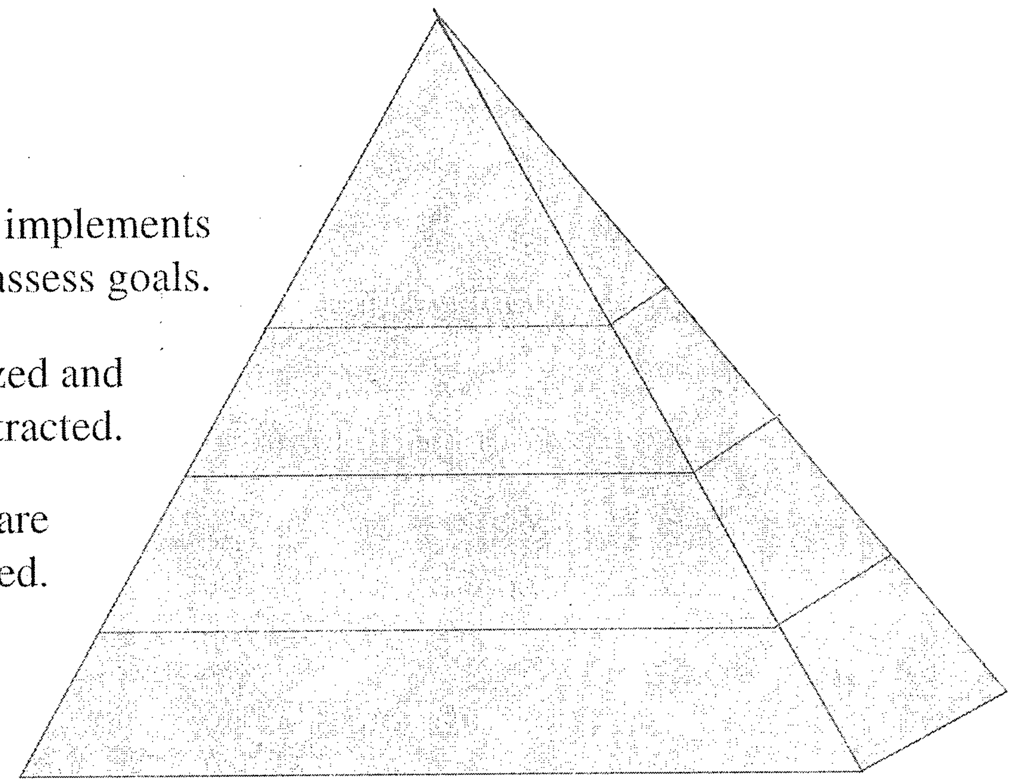
Measurements Roll-up

Management implements
process changes and reassess goals.

Data is analyzed and
useful information abstracted.

Measurements are
made and data collected.

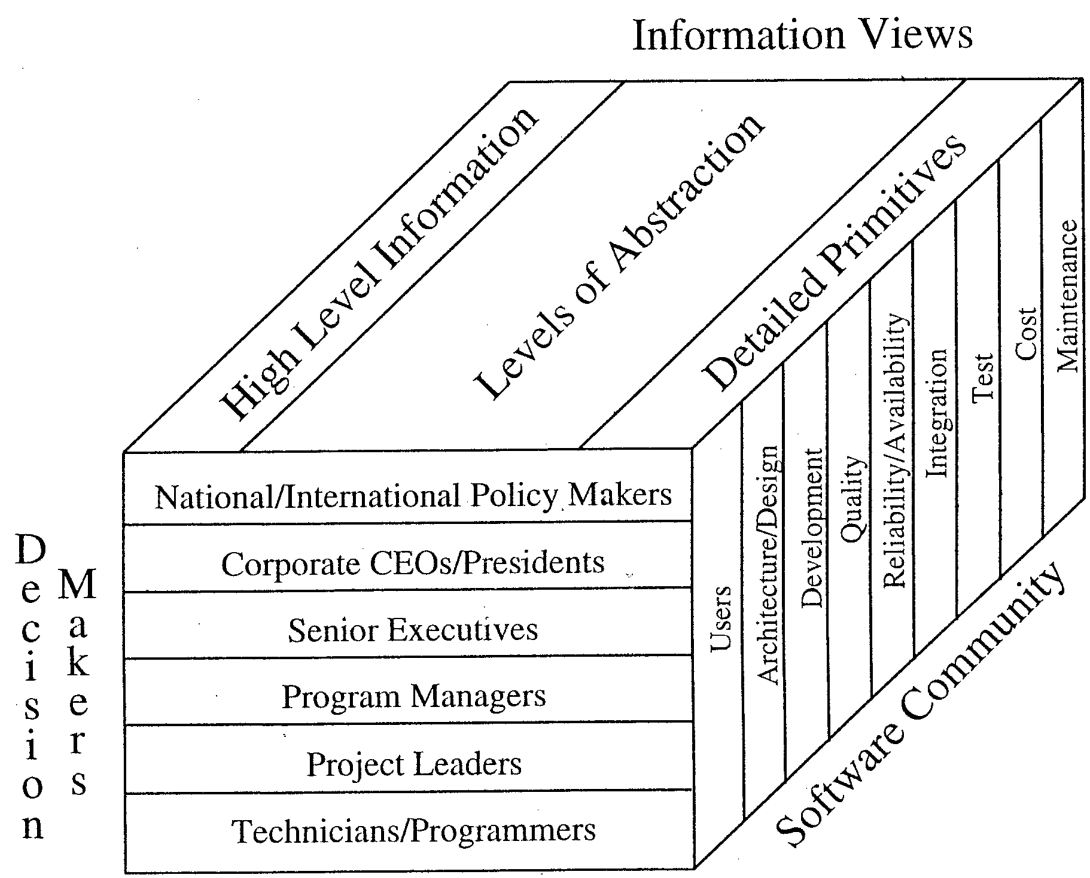
Processes followed and
primitive data generated.



Structure of a Measurement Program

The Information Hierarchy.

Software Systems Information Demographics Model



Structure of a Measurement Program

Program Based Approach

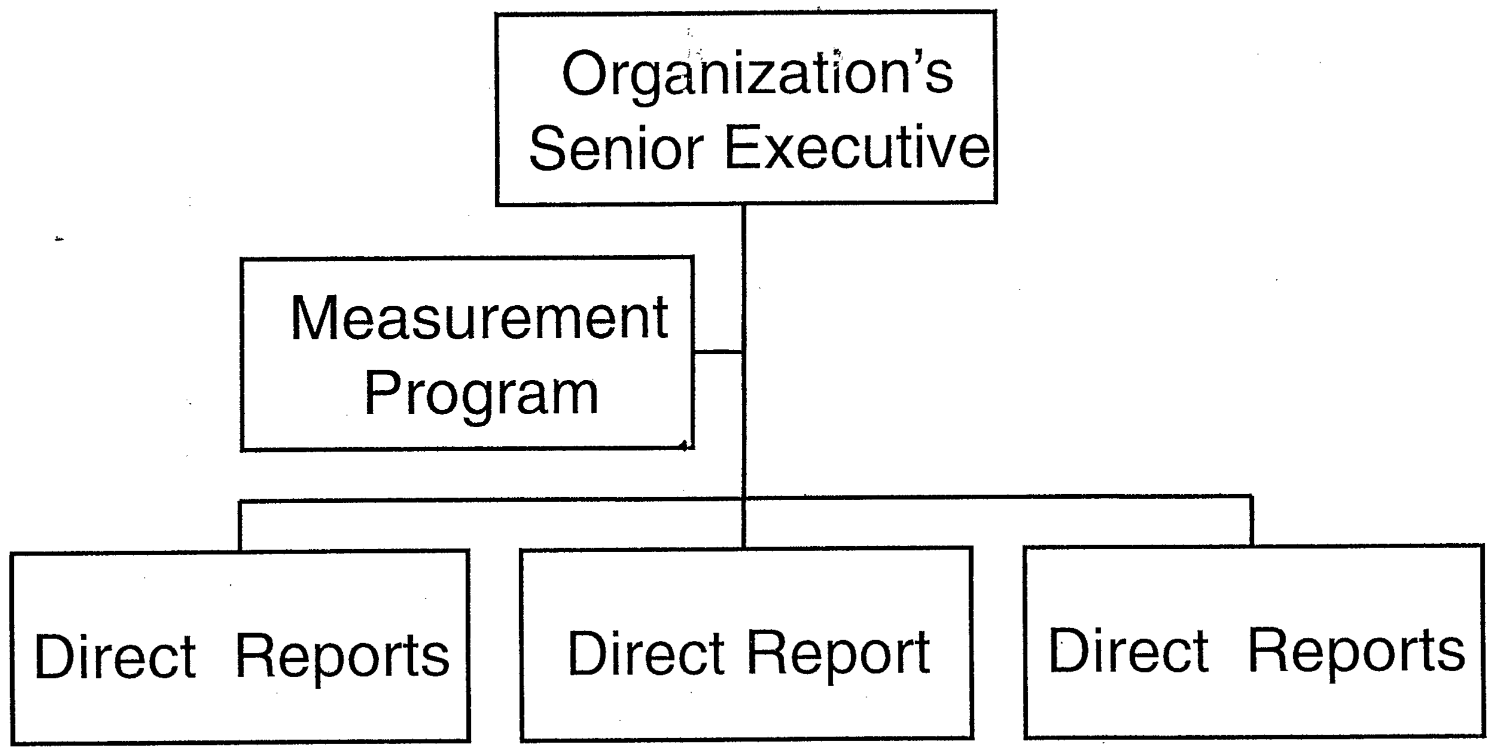
Primary Components

- Detailed Approved Program Plan
- Dedicated Resources, People, and Funding
- Oversight and Participation
  - Steering Group
  - Users Group
  - Infrastructure Team
- Training and Mentoring
- Defined Products

Structure of a Measurement Program

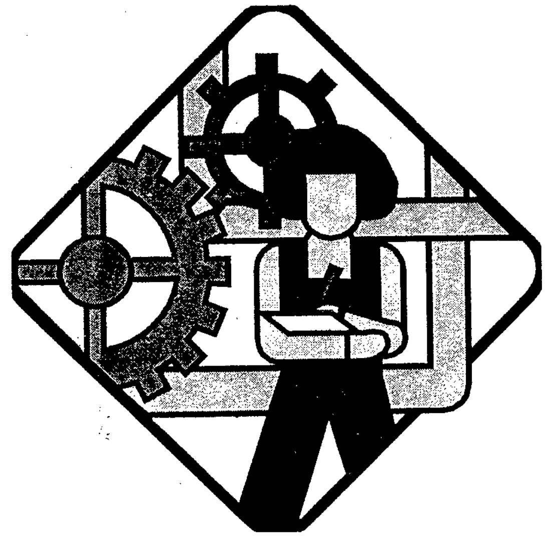
Program Based Approach

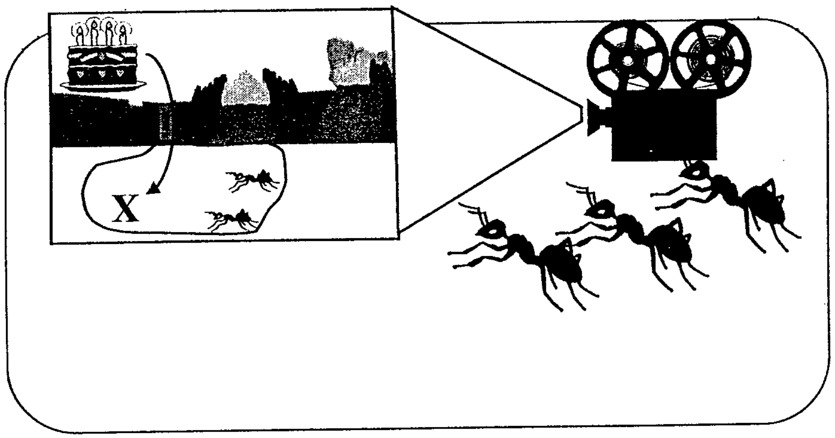
Put it were it works Best!



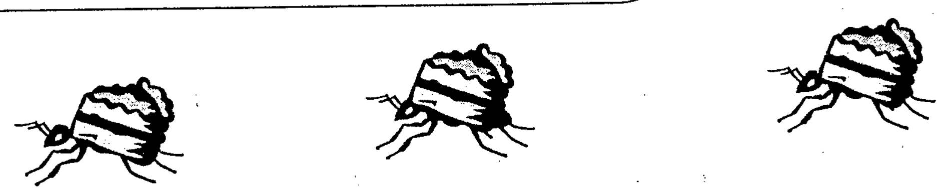
Fundamentals of Process Improvement

- How it works
- Successful Application



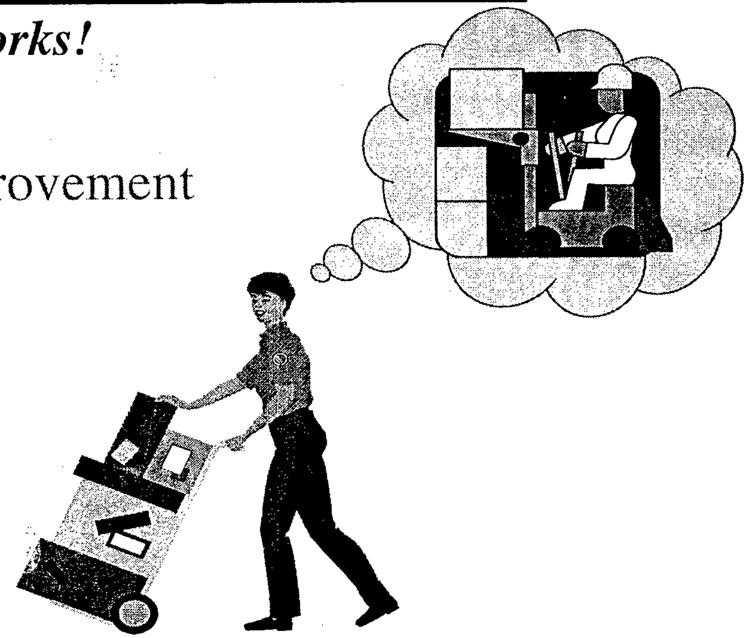


- Desired Business Results
- Power of Small Successes



Fundamentals of Process Improvement

How it Works!



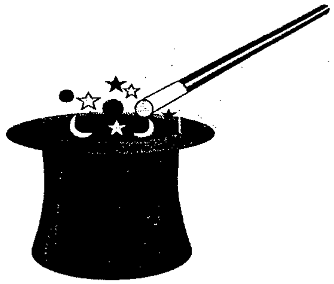
Goals of Improvement

- Increase efficiency
- Increase quality
- Increase positive cash flow

- Decrease downtime
- Decrease waste
- Decrease rework

Fundamentals of Process Improvement

How it Works!



Magic or Rocket Science?

The underlying methods of modern measurement methods are descendent from the Apollo Moon Landing Program.

Man on the Moon Program

- Determine your current coordinates.
- Determine where you want to go.
- Plot a course and speed.
- Navigate the course and adjust trajectory.
- Periodically re-calculate position.
- Rest course and speed.
- Continue on to destination.

Improvement Program

- Measure project's current status.
- Set a business objective.
- Define an Action Plan.
- Follow the Action Plan as updated for business changes.
- Measure progress and establish updated baselines.
- Modify original action plan to meet business change needs.
- Follow revised action plan until business goals are met.

Goal, Question, Measurement Paradigm

Fundamentals of Process Improvement

How it Works!

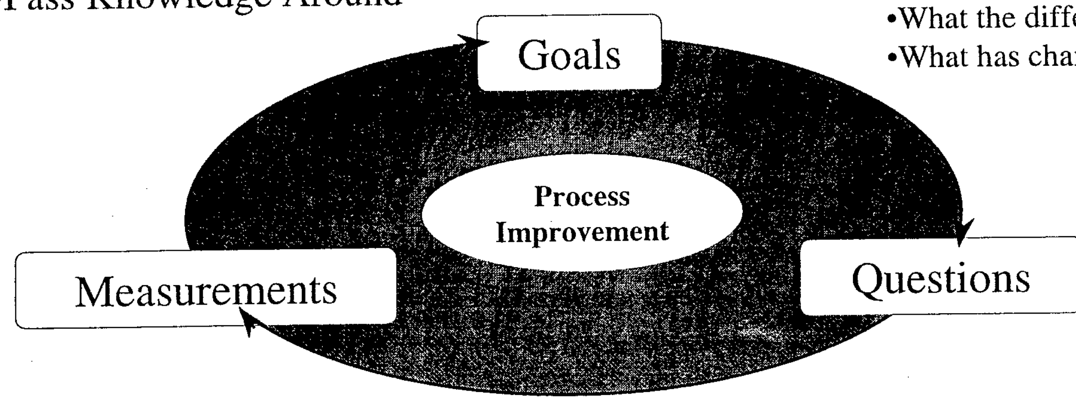
The Process Improvement Process

Figure 2

- Keep the Process Simple
  - Make Sure the Process is Repeatable
    - Pass Knowledge Around

Objective

Define simple high level objectives for the Organization based on:
- Where we are
- Where we want to go
- What the difference is
- What has changed



Measurements

Record how well the Process is working and pass the information back to refine organizational Objective and objectives.

Questions

Request information systematically to influence the implementation of most efficient procedures and documented improvements

Fundamentals of Process Improvement

How it Works!

Senior Executive

Identifies  **a Goal**

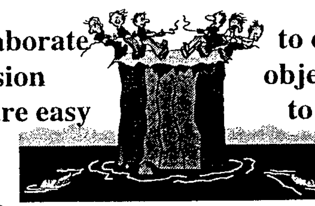
GOAL

Specific achievement to be accomplished in a given period of time.

Example:

Reduce Unit Cost by 5% over 6 months.

Direct Reports

collaborate division that are easy  **to derive objectives to digest.**

QUESTIONS

Questions identify Information needed to to make individual decisions in a cooperative environment.

Examples:

How can we reduce rework by 50%?

How can we share resources to reduce Capital Cost by 15%?

How can we increase individual productivity by 05%?

Managers Implement Technical/Management tasks, promote quality, ensure progress, and report effectiveness.

MEASUREMENTS

Specific tasks implemented, with success criteria, that can be measured and reported.

Examples:

Customer needs robustly defined, reducing requirements variance by 25%.

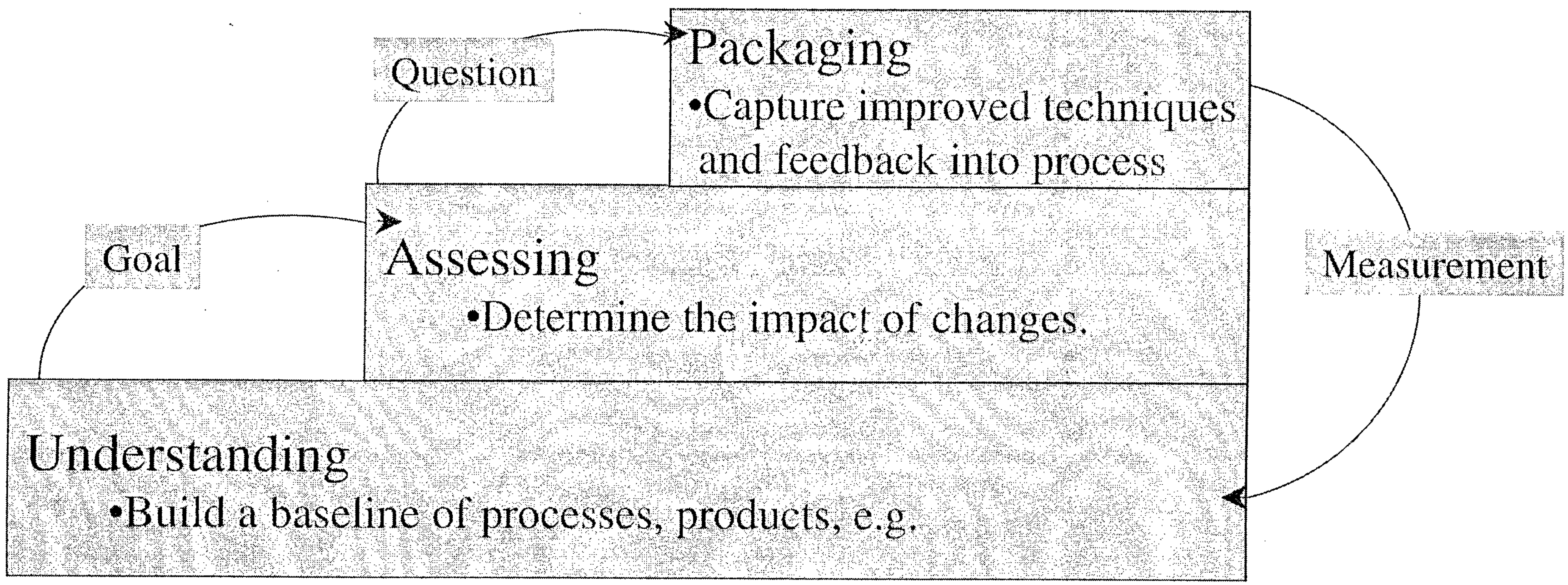
Work flow planning implemented, freeing up LAN capacity by 30%.

Application Resource Management monitors technology advancements and initiates new training.

Fundamentals of Process Improvement

A Successful Application of the Basics!

An Application of GQM



NASA's Software Engineering Library's
Three Phase Approach to Process Improvement
SEL-95-102 Software Process Improvement Guide Book

Fundamentals of Process Improvement

A Successful Application of the Basics!

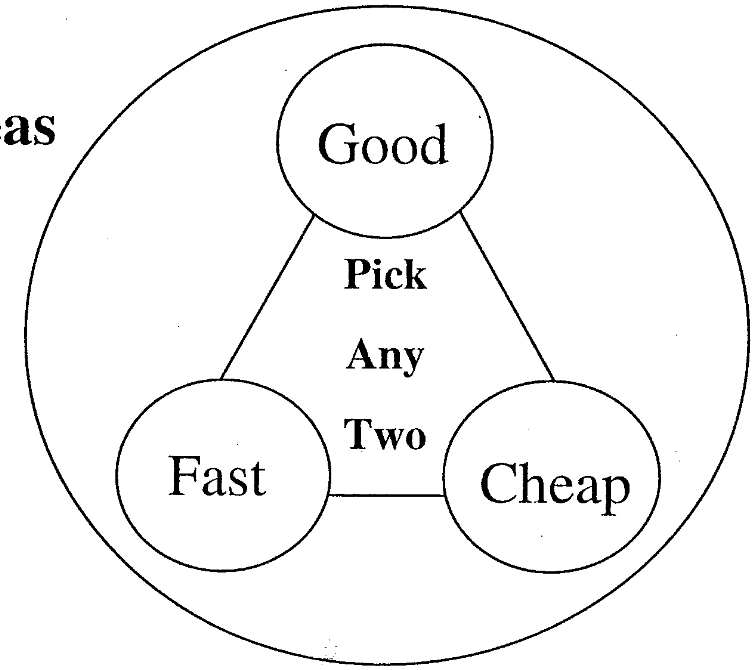
- Understanding Phase
  - The organization's process and products are characterized and high level goals for improvement are identified.
- Accessing Phase
  - Specific objectives for improvement are set, one or more changes are introduced and progress towards established goals measured.
- Packaging Phase
  - Changes that have produced satisfactory results and shown measurable improvement are institutionalized into the organization's process.

SEL-95-102 Software Process Improvement Guide Book

Fundamentals of Process Improvement

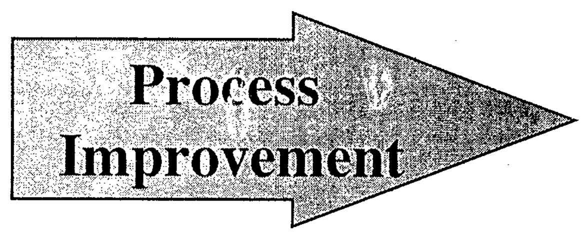
Desired Business Results!

Selecting the Improvement Areas



Today's Process

Good
Fast
Cheap

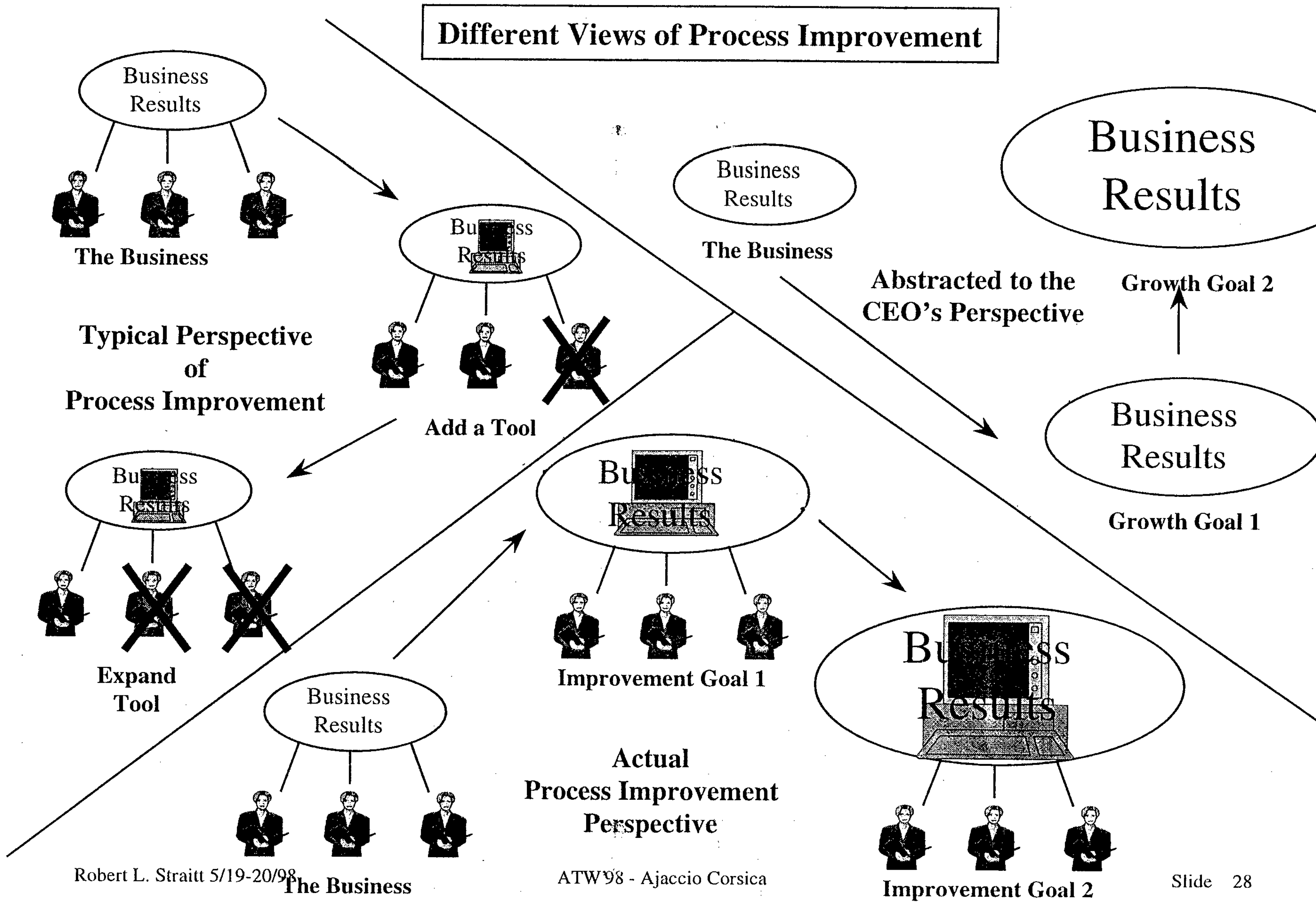


Improved Process

Better
Faster
Cheaper

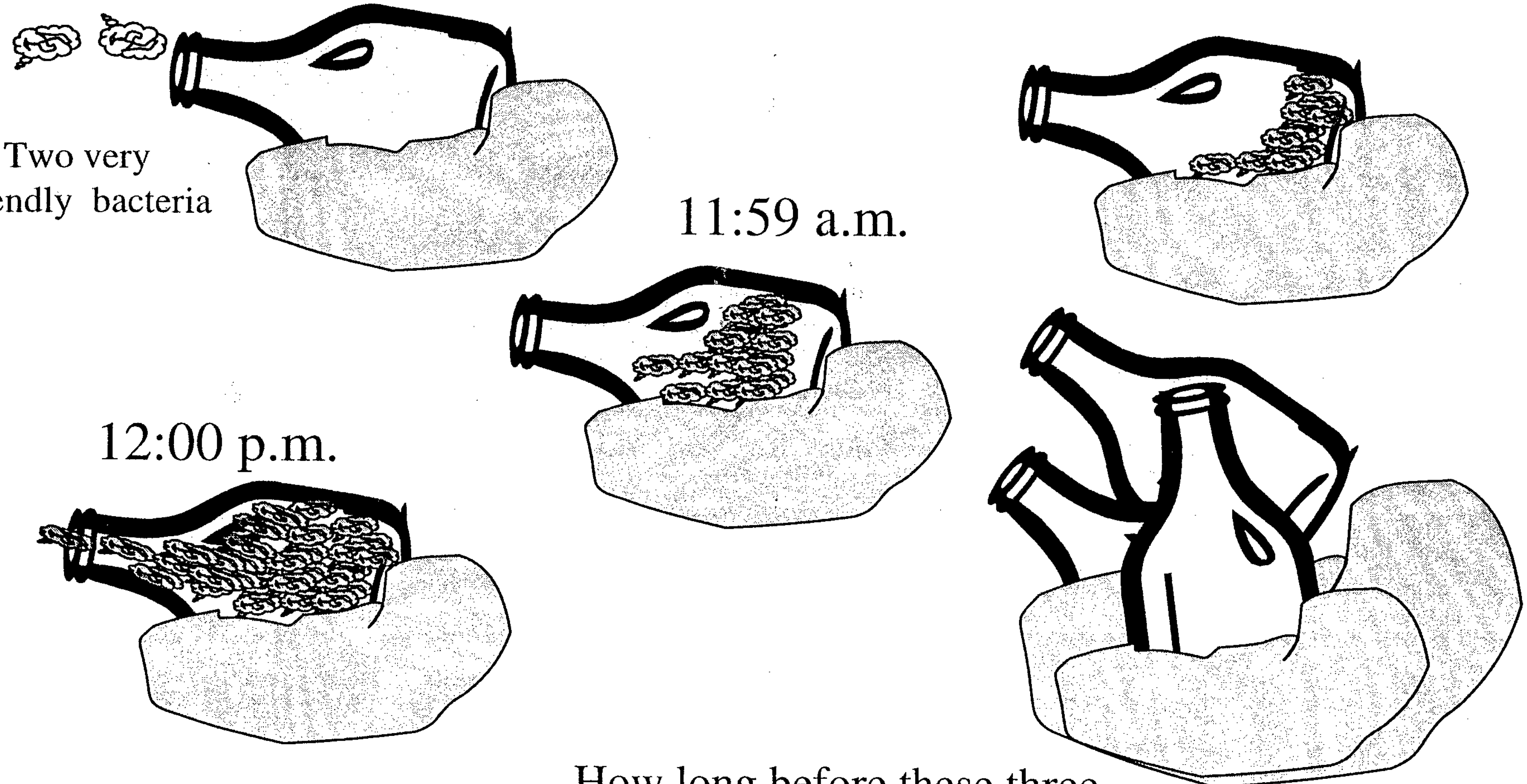
Fundamentals of Process Improvement

Desired Business Results!

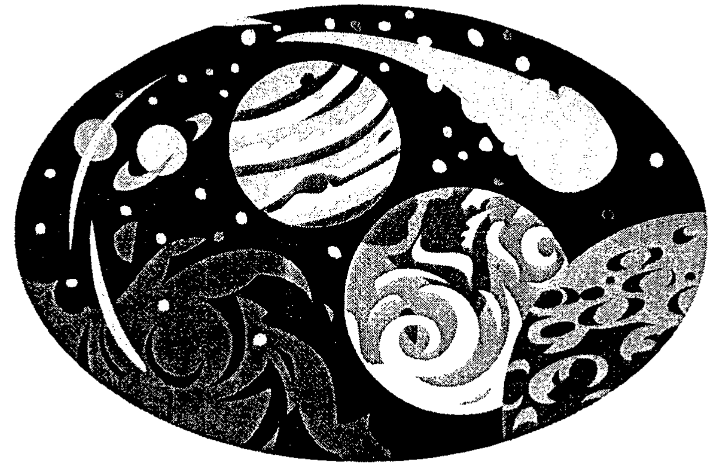


Fundamentals of Process Improvement

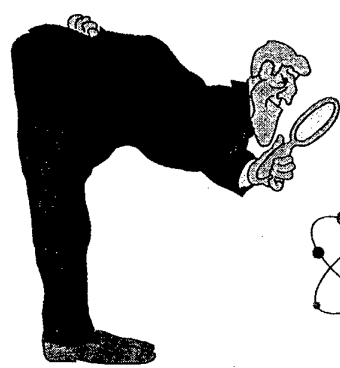
The power of small successes!

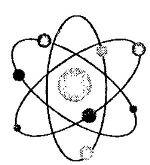


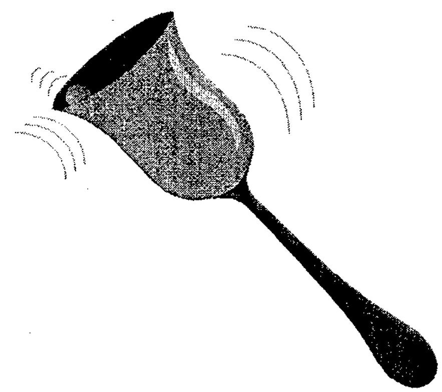
Integrating Measurements with Process



- An Integrated Model
- Communication Flows
- Matching Measures to Processes







Integrating Measurements with Process

An Integrated Model

Measurements

Indicate how you are progressing towards your goals.

Indicate areas where improvement resources can help practitioners most

Provide supporting information for technical and management decisions

Promotes active participation in improvement and quality activities.

Process Improvement

Resolves identified issues.

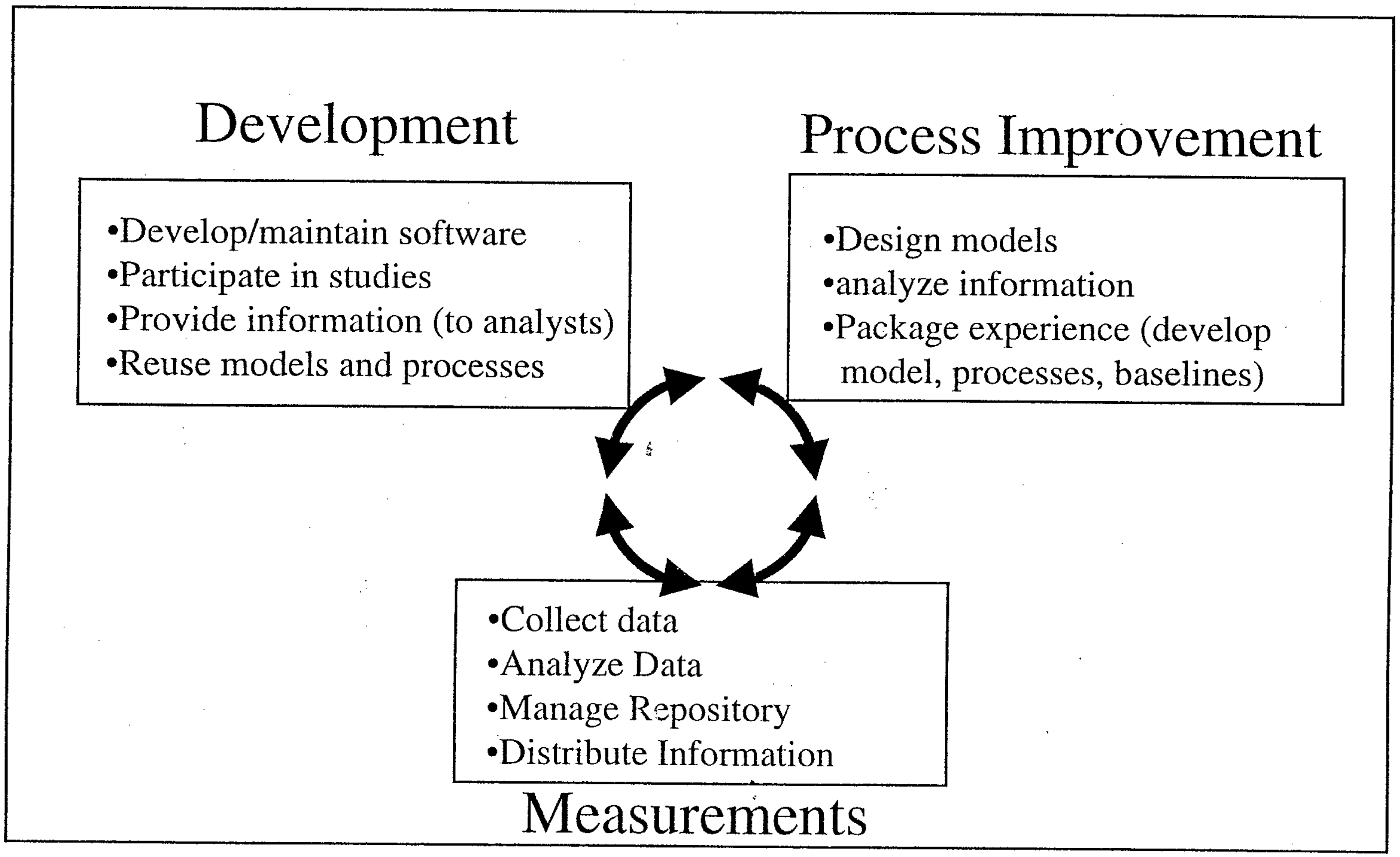
Develops action plans to meet improvement goals.

Drives increases in product quality.

Increases confidence, efficiencies and overall profitability.

Integrating Measurements with Process

An Integrated Model

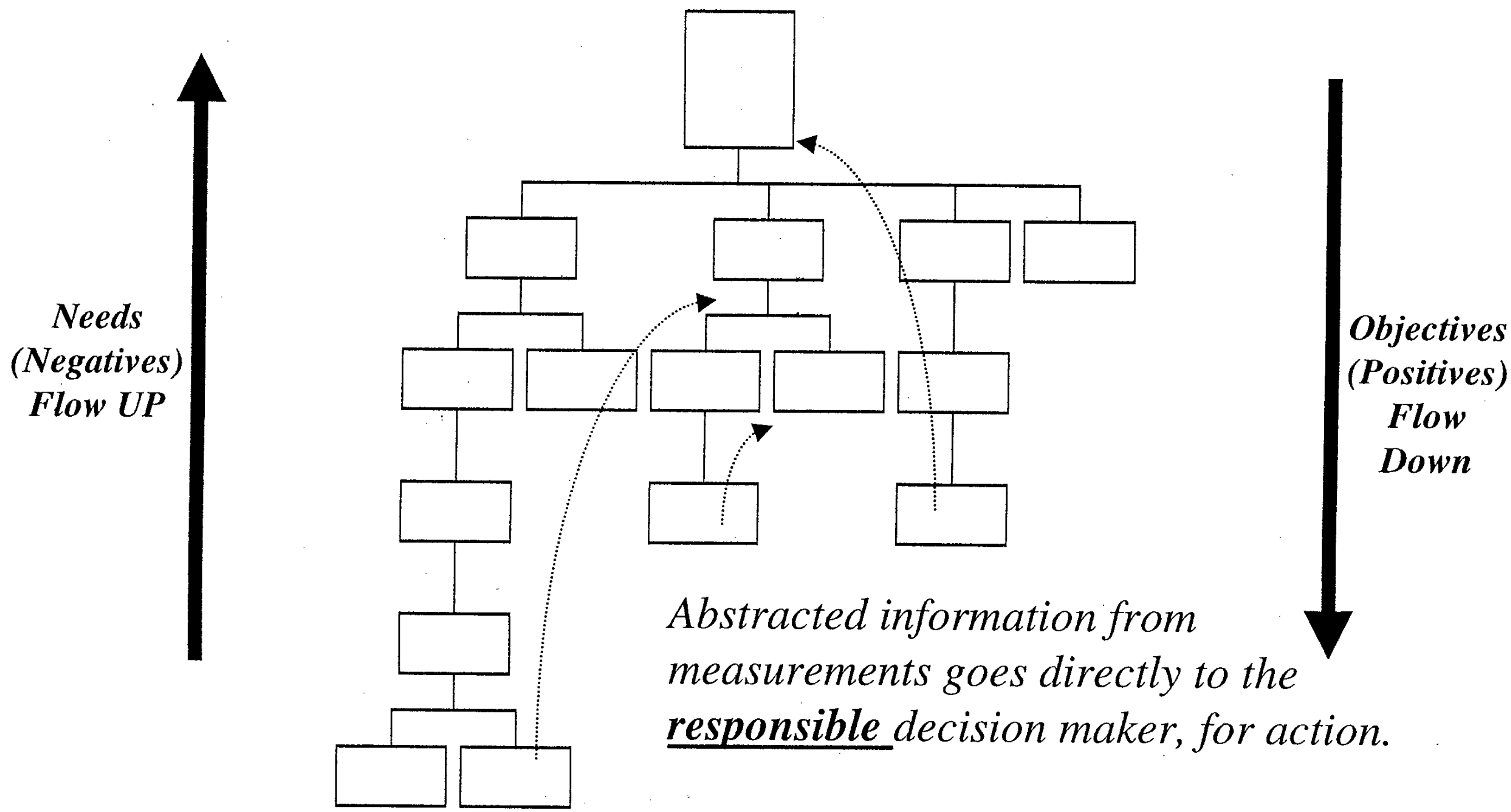


Integrating Measurements with Process

Communications Flows

Elevate problems to the highest level necessary to resolve them.

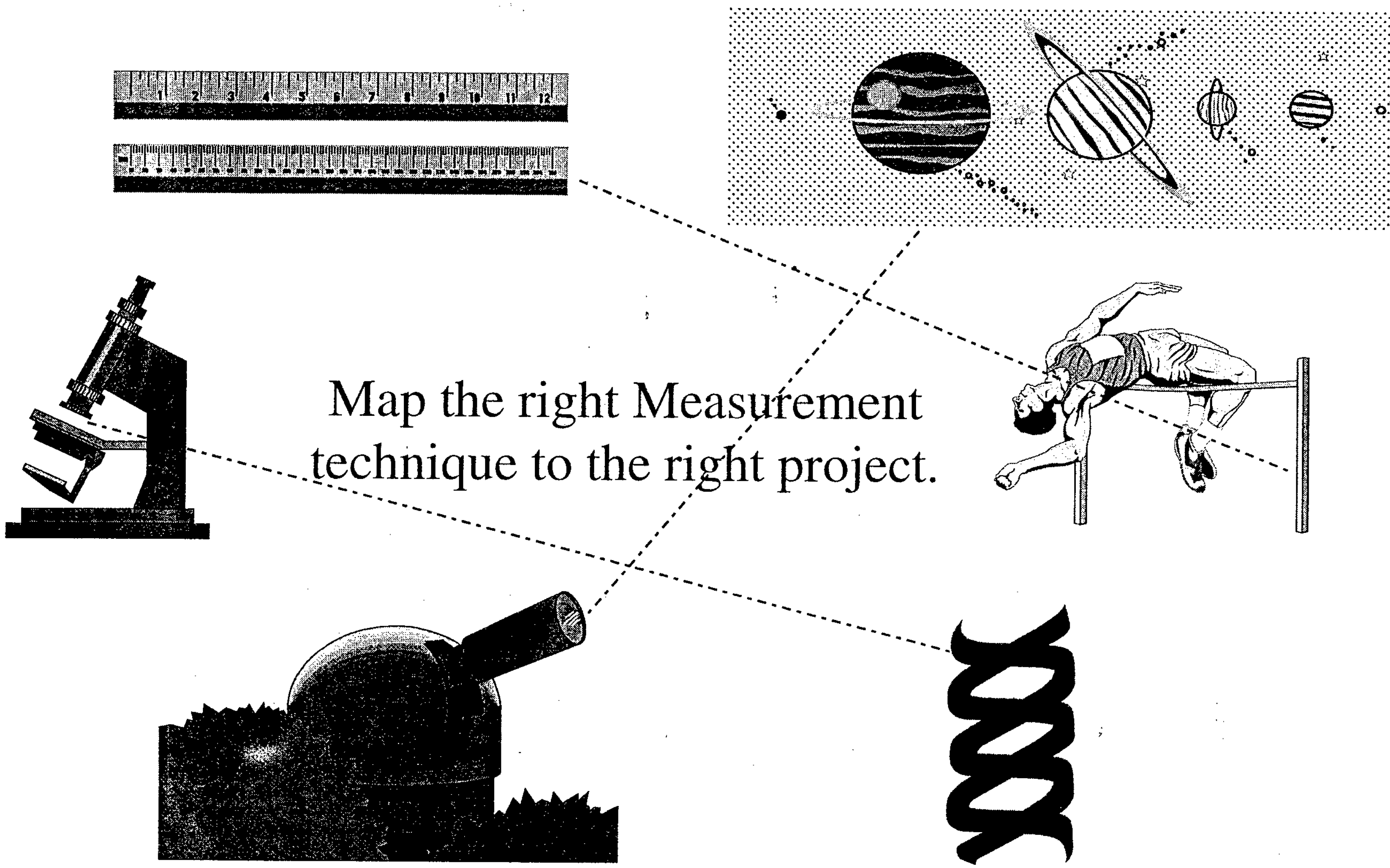
Distribute objectives and milestones down and across the organization.



Abstracted information from measurements goes directly to the ***responsible*** *decision maker, for action.*

Integrating Measurements with Process

Matching Measures to Processes



Pitfalls to Avoid

- Collecting data that has no value to anyone in the organization and will not be used.
- Using measurements to do external comparisons instead of self-improvement.
- Not controlling data accuracy or using data that has not been validated.
- Single point failures of data analysis by human error.
- Using measurements for individual performs reviews.
- Expecting automation to do all the work and not planing resource for maintenance of information.

Conclusion

- Measure where you are before you start improvements process.
- Plan goals and paths to reach those goals based on your measurement.
- Identify specific measures that show actual progression towards planed goals.
- Use and trust your measurements to guide you and make changes to your plan as necessary based on measurement feedback.

Conclusion

The right motivation makes improvement work!

I want another Apple!

Sources

- K.C. Cole, *The Universe and the Tea Cup. The Mathematics of Truth and Beauty* (Orlando, Harcourt Brace & Company, 1998).
- Watts S. Humphrey, Carnegie Mellon University, *The Capability Maturity Model. Guidelines for improving the software process.* (Addison-Wesley Publishing Company, New York, 1995).
- Morton F. Meltzer, Information, the ultimate management resource (ANACOM, New York, 1981).
- Lloyd Mosemann, II, *Improved Decision-making for Software-related Policy Making and Strategic Planning* (Presentation to U.S. Government and Industry, 1995).
- NASA, *Managers Handbook for Software Development, Revision 1, SEL-84-101* (NASA Software Engineering Library, Greenbelt, 1990).
- NASA, *Software Measurement Guidebook, Revision 1, SEL-94-102* (NASA Software Engineering Library, Greenbelt, 1995).
- NASA, *Software Process Improvement Guidebook, Revision 1, SEL-95-102* (NASA Software Engineering Library, Greenbelt, 1996).
- Robert B. Toth, et al., *The Economics of Standardization* (Standards Engineering Society, Minneapolis, 1984).
- US Air Force, Guidelines for Successful Acquisition and Management of Software Intensive Systems (USAF Software Technology Support Center, 1994)

Application of Software Metrics for Validating VHDL Descriptions of Digital Circuits

Christophe Paoli - Jean-François Santucci
SDEM – Systèmes Dynamiques, Energétiques et Mécaniques,
URA CNRS 2053
Université de Corse
BP 52 Quartier Grossetti ; 20250 Corté France
{cpaoli, santucci}@univ-corse.fr

Keywords: Cyclomatic Complexity, Control Flow Graph, Structured Testing, Software Testing, VHDL, High Level Validation.

Abstract

This paper presents a validation of VHDL behavioral descriptions at the early phase of the design of a digital system. The validation is based on a software engineering approach. After a brief introduction dealing with software testing and complexity, we will present in the first part the software testing concepts and complexity measures. In the second part we will present how these previous concepts are used in the case of VHDL descriptions.

1. INTRODUCTION

This presentation deals with validation of VHDL [1] descriptions at the early phase of a digital system. Our approach consists in generating test data from a given VHDL behavioral description. The validation is achieved by comparing the results obtained using the simulation of the VHDL description within the test data and the results which should have been obtained from the specification of the system to be designed.

The key point of such an approach is the generation of test data. In this paper we propose an original approach based on Software Testing concepts [2]. We choose such an approach because a VHDL description is a software program describing the behavior of a digital system.

The goal is to randomly generate test data. However generating random test data causes two basic problem: (i) it is necessary to define the number of test data to consider (this number is called the length of the test data in the following); (ii) it is necessary to define criteria which express the "quality" requirements that the test data have to fulfill.

To solve these problems we are concerned with testing techniques developed in the field of Software Engineering [3]. This interest is motivated by the fact that behavioral hardware languages such as VHDL and conventional languages such as C or ADA are supported by common concepts.

Having selected criteria from the field of Software Testing allowing the two aforementioned problems to be solved, we are in the phase of studying how such criteria could be measured and applied to VHDL behavioral descriptions.

In order to find criteria which could estimate the length of test data and express the quality of test data, we have been concerned with two kinds of techniques: (i) the computation of cyclomatic complexity metric (McCabe metrics [4]) and (ii) the application of coverage-based metrics [3].

The McCabe metric is based on a graphical representation of the control part of the software being tested. McCabe defined a cyclomatic number of a graph associated with the control part of software. This number represents the number of linearly independent paths of the graph. He proved that the cyclomatic number represents the minimum number of test data to be generated in order to test the control part of software.

In order to evaluate the quality of test data, conventional software testing criteria are used. These criteria correspond to coverage based metrics [5].

The first part of the paper will deal with Software Testing concepts. We will present in detail the cyclomatic complexity concept and the coverage based criteria. This part will be illustrated by pedagogical examples.

In the second part we will present how these previous concepts are used in the case of VHDL descriptions.

The last part is dedicated to a brief overview of future work we envision to perform.

2. SOFTWARE TESTING

Software testing is the process of executing software and comparing the observed behavior to the desired behavior. The major goal of the software testing is to discover errors in the software [6], with a secondary goal of giving confidence in the software when testing results are good. The quality of the testing method is then evaluated.

Complexity measure identifies software that is error-prone, hard to understand, hard to modify and hard to test. It so identifies operational steps to help control software, for example splitting complex modules into several simpler ones, or indicating the amount of testing.

There is a strong connection between complexity and testing. As a matter of fact, numerous studies and general industry experience have shown that the cyclomatic complexity measure correlates with errors.

2.1. CYCLOMATIC COMPLEXITY

Control flow graphs describe the structure of software modules. The definition of a module is language dependent. In general, it is a unit of code with a single entry point and a single exit point. Each flow graph consists of nodes and edges. The nodes represent computational statements or expressions; they may represent more than one line of code. Edges represent transfer of control between nodes.

Cyclomatic complexity [4] measures the amount of decision in a single software module. It is also known as v(G), where v refers to the cyclomatic number in graph theory and G indicates that the complexity is a function of the graph.

Method of calculation:

Given a module, whose flow graph has **e** edges and **n** nodes, its v(G) is:

$$v(G) = e - n + 2$$

Considering a set of several paths gives a matrix in which columns corresponds to edges and rows correspond to paths. From linear algebra, it is know that each matrix has a unique rank (number of linearly independent rows) that is less than or equal to the number of columns. This means that no matter how many number of possible paths are added to the matrix, the rank can never exceed the number of edges in the graph. In fact the maximum value of a rank is exactly v(G). A minimal set of vectors (paths) with maximum rank is known as a basis. A basis can also be described as a linearly independent set of vectors that generate all vectors in the space by linear combination. So a basis is the minimum number of paths that should be tested. Therefore v(G) is the number of paths in any independent set of paths that generate all possible paths by linear combination. Given any set of paths, it is possible to determine the rank by doing Gaussian Elimination on associated matrix. The rank is the number of non-zero rows once elimination is complete. If no rows are driven to zero during the elimination, the original paths are linearly independent. If the rank is equal to v(G), the original set of paths generates all paths by linear combination. If both conditions hold, the original set of paths is a basis for the flow graph.

2.2 STRUCTURED TESTING

Basis path testing or structured testing use v(G) to guide the testing process. Structured testing is more theoretically rigorous and more effective at detecting errors in practice than other common test coverage criteria such as statement coverage and branch coverage [7].

Structured testing as presented in this sub-section applies to individual software modules. It is simply stated: " Test a basis set of paths through the control flow graph of each module ". This means that any additional path can be expressed as a linear combination of paths that have been tested.

This criterion establishes a complexity number, v(G), of test paths that have two critical properties:

- A test set of v(G) paths can be realised.
- Testing beyond v(G) independent paths is redundantly exercising linear combinations of basis paths.

Therefore the minimum number of tests required to satisfy the structured testing is exactly v(G).

Note that structured testing criterion measures the quality of testing, providing a way to determine whether testing is complete. It is not a procedure to identify test cases or generate test data inputs. The independent test paths can be identified by the baseline method described below.

The baseline method is a technique for identifying a set of control paths to satisfy the structured testing criterion.

The idea is to start with a baseline path (it is the first path selected by the tester), then vary exactly one decision outcome to generate each successive path until all decision outcomes have been varied. A basis is then generated.

The selection of this baseline path is somewhat arbitrary. The key point is to pick a representative function rather an exceptional condition.

3. OUR APPROACH FOR VHDL BEHAVIORAL MODEL

In this section we describe in detail how the previous concepts are used for the validation of VHDL behavioral descriptions. The methodology we are developing is illustrated on an example of VHDL description.

3.1. THE REGISTER'S EXAMPLE

In this sub-section VHDL behavioral model of the register is presented. The nodes corresponding with lines of code are noted from *A* to *T* in the register's model. It is shown in figure 1.

```
    Entity Register IS
        Port (DI . IN vlbit_1d(1 TO 8) ;
                STRB, DS1, NDS2 : IN vlbit ;
                DO : OUT vlbit_1d(1 TO 8)) ;
        END Register

    Architecture behavior of Register IS
        SIGNAL reg : vlbi_1d(1 TO 8);
        SIGNAL enbld : vlbit ;
A,B,C,R,S       BEGIN

D       strobe: PROCESS (STRB)
        Begin
E,F             If (STRB =1) Then reg <= DI;
G        End If;
H               End PROCESS strobe;
I       enable : PROCESS (DS1,NDS2)
```

```
        Begin
J         enlbd <= DS1 AND NOT (NDS2);
K       END PROCESS enable;

L       output : PROCESS (reg,enbld)
        Begin
M,N      If (enbld=1) then DO<=reg
O        Else DO <= 11111111;
P        end If
Q       END PROCESS output;

T       END Behavior
```

Figure 1. Register's Model

3.2. *THE REGISTER'S FLOW GRAPH*

We present in this sub-section the control flow graph of the register's model with its edges numbered from 1 to 24, and its nodes noted from *A* to *T*. For that example we have found a complexity equal to 6 (24 edges minus 20 plus 2). The control flow graph is shown in figure 2.

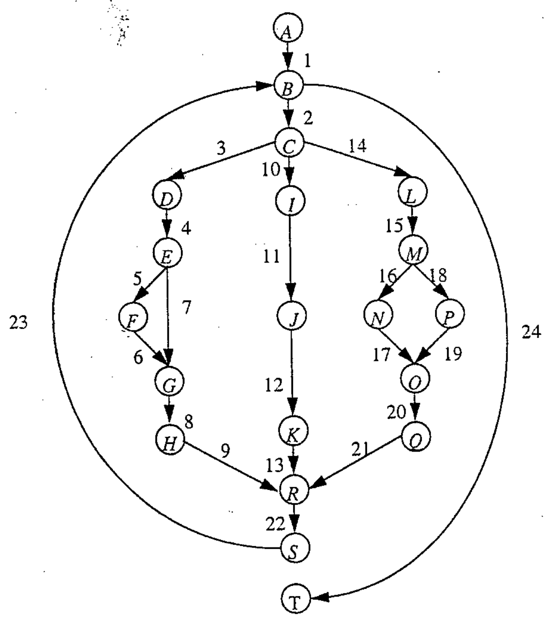


Figure 2. Control Flow Graph for the Register's Model

3.3. *THE BASIS TEST PATHS*

Since complexity is equal to 6, a test set of 6 paths should be obtained. These paths are noted from P1 to P6. A redundant test (it is equal to P2+P3+P5-2*P6) is also generated. It is the path corresponding to the initialisation phase (all the processes are active). It is noted Pinit. In figure 3, a signal is said active if the value of the signal during the previous VHDL simulation cycle is different from the actual one's.

<u>*Test Path 1 :*</u> *1.2.3.4.5.6.8.9.22.23.24*
Node B : STRB or DS1 or NDS2 or reg or enbld is active : TRUE
  Node C : STRB active : TRUE
    DS1 or NDS2 active : FALSE
    reg or enbld active : FALSE
  Node E : STRB = 1 : TRUE
Node B : STRB or DS1 or NDS2 or reg or enbld is active : FALSE

<u>*Test Path 2 :*</u> *1.2.3.4.7.8.9.22.23.24*
Node B : STRB or DS1 or NDS2 or reg or enbld is active : TRUE
  Node C : STRB active : TRUE
    DS1 or NDS2 active : FALSE
    reg or enbld active : FALSE
  Node E : STRB = 1 : FALSE
Node B : STRB or DS1 or NDS2 or reg or enbld is active : FALSE

<u>*Test Path 3 :*</u> *1.2.10.11.12.13.22.23.24*
Node B : STRB or DS1 or NDS2 or reg or enbld is active : TRUE
  Node C : STRB active : FALSE
    DS1 or NDS2 active : TRUE
    reg or enbld active : FALSE
Node B : STRB or DS1 or NDS2 or reg or enbld is active : FALSE

<u>*Test Path 4 :*</u> *1.2.14.15.16.17.20.21.22.23.24*
Node B : STRB or DS1 or NDS2 or reg or enbld is active : TRUE
  Node C : STRB active : FALSE
    DS1 or NDS2 active : FALSE
    reg or enbld active : TRUE
  Node M : enbld = 1 : TRUE
Node B : STRB or DS1 or NDS2 or reg or enbld is active : FALSE

<u>*Test Path 5 :*</u> *1.2.14.15.18.19.20.21.22.23.24*
Node B : STRB or DS1 or NDS2 or reg or enbld is active : TRUE
  Node C : STRB active : FALSE
    DS1 or NDS2 active : FALSE
    reg or enbld active : TRUE
  Node M : enbld = 1 : FALSE
Node B : STRB or DS1 or NDS2 or reg or enbld is active : FALSE

<u>*Test Path 6 :*</u> *1.24*
Node B : STRB or DS1 or NDS2 or reg or enbld is active : FALSE

<u>*Test Path init :*</u> *1.2.3.4.7.8.9.22.23.2.10.11.12.13.22.23 .2.14.15.18.19.20.21.22.23.24*
Node B : STRB or DS1 or NDS2 or reg or enbld is active : TRUE
  Node C : STRB active : TRUE
    DS1 or NDS2 active : FALSE
    reg or enbld active : FALSE
    Node E : STRB = 1 : FALSE
Node B : STRB or DS1 or NDS2 or reg or enbld is active : TRUE
  Node C : STRB active : FALSE

DS1 or NDS2 active : TRUE
reg or enbld active : FALSE
Node B : STRB or DS1 or NDS2 or reg or enbld is active : TRUE
Node C : STRB active : FALSE
DS1 or NDS2 active : FALSE
reg or enbld active : TRUE
Node M : enbld = 1 : FALSE
Node B : STRB or DS1 or NDS2 or reg or enbld is active : FALSE

Figure 3. Basis test of Paths for the Register's Model

3.4. *THE MATRIX OF EDGE INCIDENCE*

As seeing in section 2, a set of several paths can be described by a matrix, which shows the number of time each edges is executed along each path. The matrix is shown in figure 4.

Paths\Edges	1	2	3.4	5.6	7	8.9	10.11.12.13	14.15	16.17	18.19	20.21	22.23	24
P1	1	1	1	1	0	1	0	0	0	0	0	1	1
P2	1	1	1	0	1	1	0	0	0	0	0	1	1
P3	1	1	0	0	0	0	1	0	0	0	0	1	1
P4	1	1	0	0	0	0	0	1	1	0	1	1	1
P5	1	1	0	0	0	0	0	1	0	1	1	1	1
P6	1	0	0	0	0	0	0	0	0	0	0	0	1
Pinit	1	3	1	0	1	1	1	1	0	1	1	3	1

Figure 4. Matrix of Edge Incidence

4. CONCLUSION

We have presented in this paper how software testing concepts could be used for deriving test benches for VHDL descriptions. The approach is based on McCabe cyclomatic complexity and structured testing method.
We have chosen to describe on a VHDL behavioral description how the concepts stemming from software testing could be applied for the validation of VHDL descriptions.
We are in the phase of define the specification of a software allowing to automatically generate test data for VHDL descriptions from the previous concepts.
Our future work will deal with the following points:

- To study of software integration approach in order to deal with complex VHDL descriptions.
- To define the specification of a test benches generator using these software testing concepts.

REFERENCES

[1] VHDL, Language reference Manual, IEEE Standard 1076, 1987.

[2] E. Miller, Introduction to Software Testing Technology in Tutorial: Software Testing & Validation Techniques, Ed. E. Miller, W.E. Howden, IEEE Computer Society Press, pp.4-16.

[3] W.E. Howden, A surwey of Dynamic Analysis Method, in Tutorial: Software Testing & Validation Techniques, Ed. E. Miller W.E. Howden, IEEE Computer Society Press, pp.209-231.

[4] T.J McCabe, A Complexity Measure, IEEE Trans. Software Testing Engineering, N°2, 1976, pp.308-320.

[5] A.H. Watson, T; J McCabe, Structured Testing: A testing Methodology using the cyclomatic Complexity Metric, NIST Special Publication 500-235, August 1996.

[6] G. Myers, The Art of Software Testing, Wiley, 1989.

[7] A. Watson, Structured Testing: Analysis and extensions, Ph.D. dissertation, Princeton University.

Bottlenecks in System-On-A-Chip Designs

Visegrády, Tamás
University of New Hampshire, USA

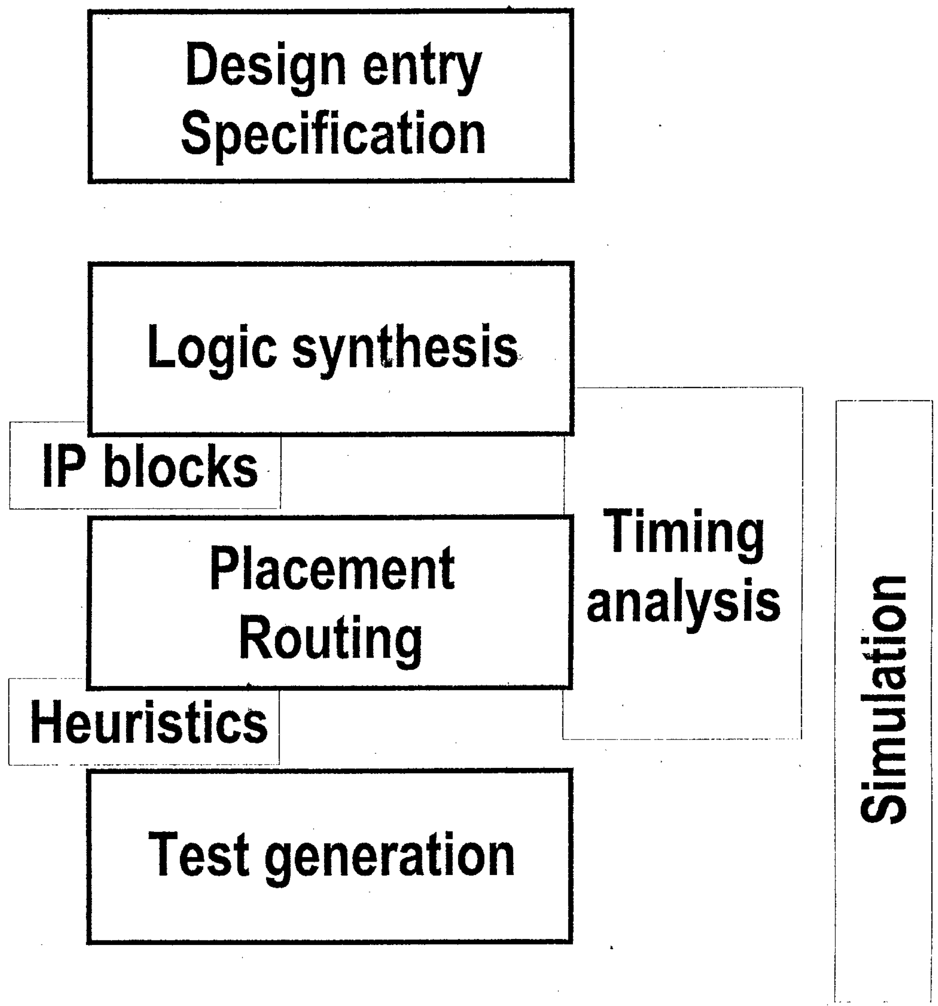
ATW'98, Ajaccio, Corsica

May 1998

Summary

- Design Process
- High-Level Synthesis
- System-On-A-Chip Designs
- Moore's Law: Good or Bad?
- Testing and Verification
- Conclusions

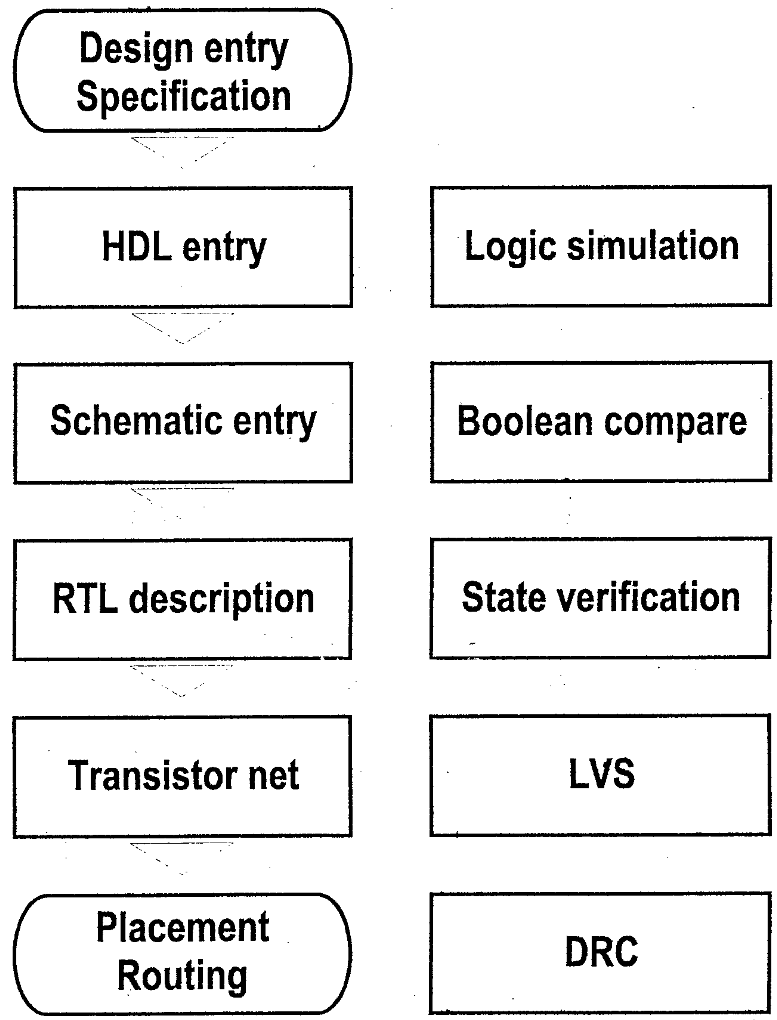
Design Process



Design Process

- Top-down problem partitioning
- Bottom-up submodules
- System specification
- Synthesis
- Realization
- Testing and verification
- Heuristic refinement steps

High-Level Synthesis



High-Level Synthesis

- Between specification and layout
- Heuristic steps
- Uses knowledge from compiler theory
- Relies on existing module base
- Commercial packages limited in scope

High-Level Synthesis: Testing and Verification

- Abstraction decrease with time
- Exponential increase in problem set complexity
- Radically different test and verify tools required in different stages

System-On-A-Chip Designs

- Single package complete systems
- Problem scope partially outside electric engineering
- Integration issues
- Lack of proper tool support
- Testing and verification problems

Moore’s Law: Good or Bad?

Target complexity

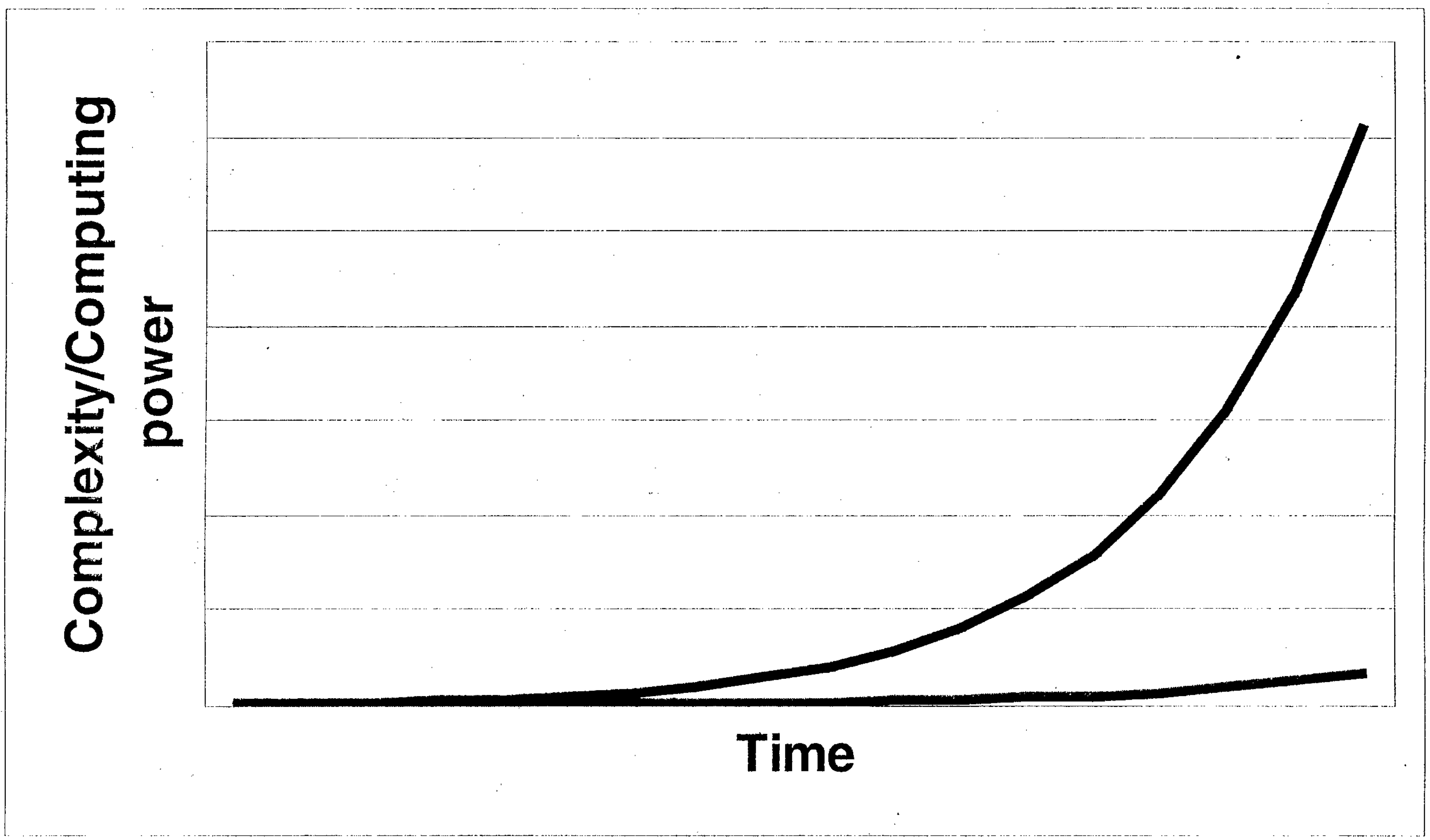
$$\frac{2^{(M^t)}}{\alpha\, M^t\, F(t)}$$

Constant

Moore's law

Machine count
(Linear, Fibonacci...)

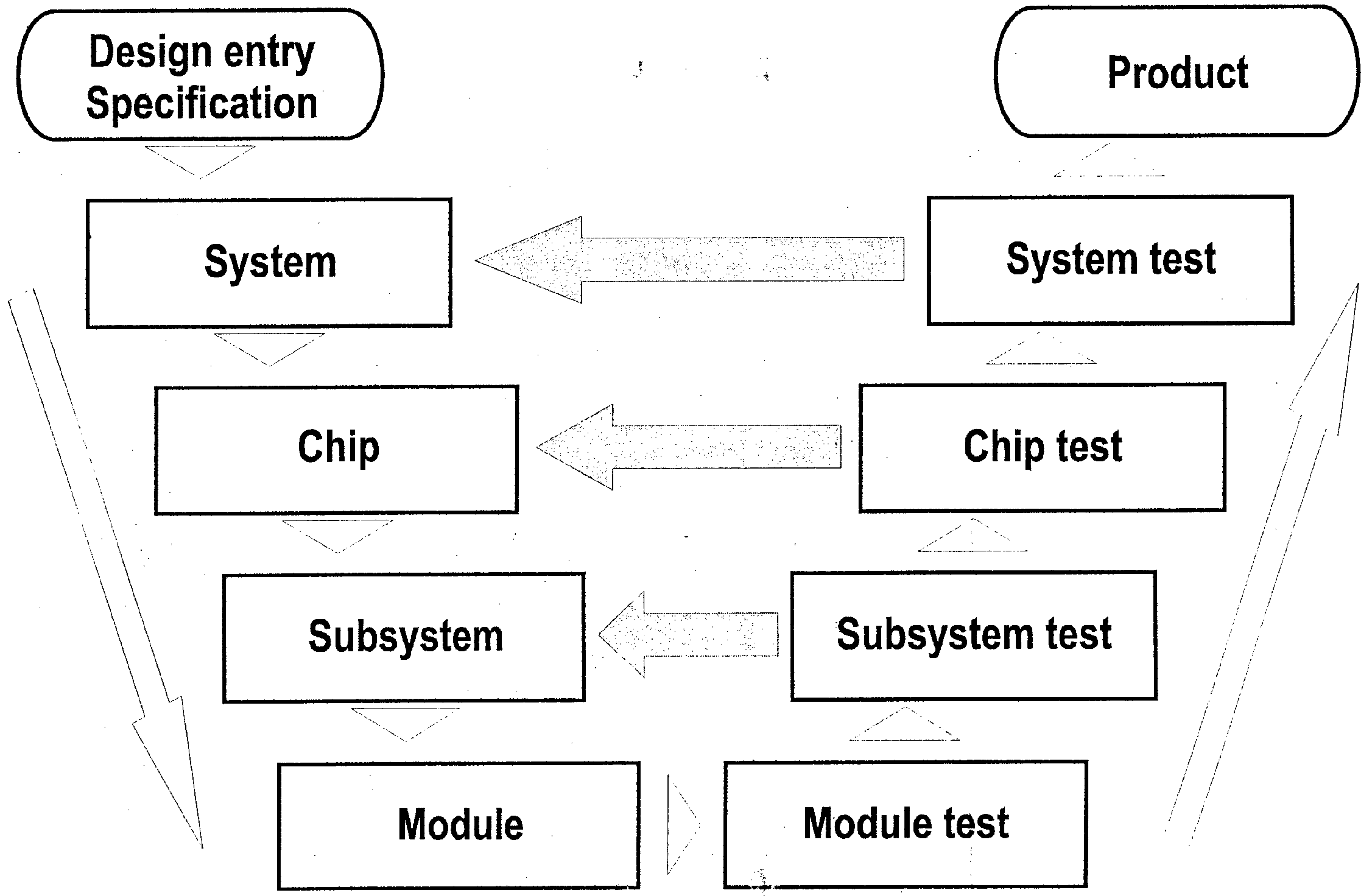
Moore's Law: Good or Bad?



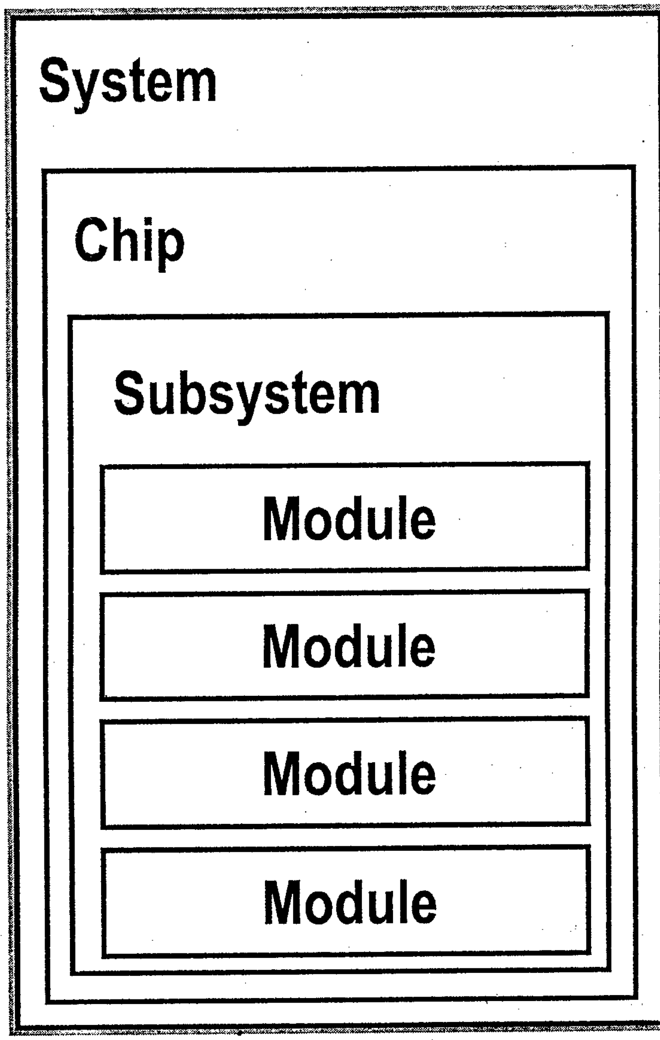
Moore's Law: Implications

- Exponential increase in target complexity (time/size)
- Slower computational power increase (time/size)
- Distributed computation has limited use for very large systems

Testing and Verification



Testing and Verification



- Local optimization only
- Lack of heuristics
- Partitioning problems
- Verification and testing differ in scope

Testing and Verification

- Mirrors reversed design flow
- Iterative process
- Non-electric feedback as part of System-On-A-Chip processes
- Problems with hierarchical T & V

Conclusions

- Increasing gap between design and verification process
- Heuristics form SOC design backbone
- Collaborative effort of EE, CS and ME specialists

Testability estimates in global data path allocation

K. Olcoz†, J.F. Tirado†, J.F. Santucci‡

†Depto. de Arquitectura de Computadores y Automática, Univ. Complutense (Spain)

‡Faculte des Sciences et Techniques, Univ. of Corsica (France)

Abstract

In this paper we present and approach to the allocation of data paths that uses scheduling information and low level area information together with testability in order to synthesize good data paths. Particular emphasis is put in developing a method for testability estimation suited for the different uses it is given during data path allocation in an environment of HLS.

Keywords: Synthesis for Testability, Synthesis for BIST, Data path allocation.

Introduction

Recent advances in VLSI technology are motivating changes in the traditional methods of design and test, that can no longer be undertaken separately. Testability, defined as the facility to generate and apply the test, is added as a new constraint to the synthesis process and design modifications are proposed to improve testability [1]. One popular Design for Testability methodology is Built-In Self-Test (BIST) design, which includes in the same chip the circuits that perform the test (**test circuits**) with the circuits that correspond to the design (**functional circuits**).

To make a data path self-testable using the pseudo-random BIST methodology, it has to be reconfigurated during test mode into a set of acyclic logic blocks, each one with a pseudo-random test pattern generation register (TPG) at each of its inputs, and a signature analyzer register (SA) at each of its outputs [2]. BIST requires reconfiguration of some functional register as a TPG or SA. A third kind of register, called BILBO [3], can be configured alternatively as a TPG and an SA. But a register cannot be configured both as a TPG and an SA simultaneously, unless it is implemented as a concurrent BILBO (CBILBO) [5], which is very expensive in terms of area and delay penalties. Hence, a **self-adjacent** register, which serves both as input and as output of a logic block, poses a problem if it has to be implemented as a CBILBO. In the most general BIST scheme, a **test path** through which test data can go from the TPGs to the SA at the output of a logic block may pass through several functional units (FUs). If two test paths share the same hardware a conflict is created, forcing the need for multiple **test sessions**.

As a result, synthesis for BIST tools generate data paths that in normal function mode execute the normal behavior, while in test mode they can be configured to perform the test of every module in a number of test sessions. High level synthesis (HLS) consists mainly of scheduling and allocation [6]. The scheduling of operations to control steps binds each operation of the behavioral description to a specific control step, determining the tradeoff between area and speed of the circuit. Allocation consists of generating a data path of minimum area starting from the scheduled description of the behavior. It involves allocation of operations to functional units (or modules), results to registers and data transfers to connections. For a given operation there are usually FUs of different types in the library (e.g. for the addition there are ripple-carry adders, carry look-

ahead adders, ALUs performing addition and subtraction and so on), so module allocation consists of selection of the type of FU and of the instance. When both tasks are split, the later task is called binding, while the former is module allocation.

The synthesis process involves navigation through the space of possible designs with the specified behavior (e.g. the **design space**), making appropriate tradeoffs, until the best solution satisfying the constraints is reached. All the data paths in the design space that can be tested and meet the area constraint are valid solutions. Selection of one of them is usually done by the user.

The rest of the paper introduces the approach to data path allocation and the testability estimations needed for it. Finally some results and conclusions are presented.

1. Global allocation scheme

HLS tasks are interdependent, so the more the relationships between them are taken into account, the better the designs synthesized. Not only the allocation sub tasks are related but also the scheduling of operations to control steps is closely related to the type and number of FUs needed to implement the operations. To bear in mind this dependence, some systems perform scheduling together with module allocation. Thus, they determine simultaneously the control step and the type of module each operation will be assigned to. Then, they perform binding as the mapping of operation and variables to particular instances of the modules and registers.

The main drawback of this approach is that module allocation is performed without regard to module binding, that determines the RTL structure of the data path and the interconnect area, that is, the area of multiplexers and wires. Measures of the area of real circuits have shown that interconnect area can take up substantial circuit area, so that performing allocation without regard to binding can lead to non minimum data paths. Thus, the smaller design is not always made up of the minimum set of modules and registers. Some extra modules and/or registers can be included if that decreases interconnect area.

In our approach the relationships between module allocation and the other two sub problems (scheduling and binding) are both borne in mind. So, scheduling performs a tentative module allocation [7]. Then, the final module allocation and binding are performed starting with this initial module allocation. The features related to the implementation of all the different types of modules are stored in a library of available modules. It is used during scheduling to determine which module, that is, which type, is most suitable for each node. Further, for each node, a list of all the different implementations available for the operation of the node (e.g. all the types of adder if the operation is an addition) is provided by the scheduler to help during allocation. The types in the list are sorted according to their suitability to implement the node once the schedule and the cycle time have been fixed. When the node has to be allocated to a module this information is useful.

To summarize, the inputs to the allocation algorithm are a scheduled CDFG and the tentative module allocation for each operation (that is, the order in which the types of module have to be tried, and the number of modules of each type for the first solution) that are provided by the scheduler. Information of the available types of modules and registers is provided as a library of modules. It also holds low level (layout) information, needed for the precise estimation of module and interconnect area. The library of modules used throughout the paper has been generated using CADENCE-ES2 1 um technology.

1.1. Allocation algorithm

We use a branch and bound algorithm so that it searches the portion of the design space allowed by the bounding function and finds a number of solutions with different testability and area figures [8], [9], [10]. One of the solutions reached must be chosen as the final design.

In order to obtain each solution we split the allocation problem into a set of sub problems, one for each **atomic allocation** of an operation or a result. Then we solve each one of them sequentially beginning from the first control step. Within a control step, operations scheduled to begin at the control step are chosen before variables generated in the control step.

The algorithm for the atomic allocation starts from a partially allocated CDFG and a partial design (that is, the set of FUs, registers and connections that has been allocated up to the moment). An operation is allocated to an instance of an FU while a variable is allocated to an instance of a register. Data transfers are assigned to interconnections after each allocation. Interconnections are reused or added if necessary.

There are usually several alternatives for the allocation of any operation or result. In the case of allocation of an operation the candidates are, on the one hand, the functional units in the partial design that are able to perform the operation of the node and that are idle in the control step considered. Moreover, another alternative is adding a new functional unit to the partial design.

The testability and area added to the former partial design by selection of every candidate are estimated. The area increment of the candidates which are already in the partial design, if any, comes from the creation of connections and also from the growing in size of the multiplexers at the inputs of the candidate. Area increment caused by addition of a new element to the partial design is the sum of the size of the element and the creation of the connections to its inputs [9].

Then, testability and area estimations are compared to check values given by the bounding functions. Branches producing new partial designs with testability or area estimations not allowed are discarded to save exploration time.

If the bounding function works correctly, we are sure that branch and bound algorithms find the optimal solution when the search time is not limited. The search time is determined by the order in which candidates are explored. If the best options are taken in the first places, the best design is reached in a small amount of exploration time. Some heuristics try to predict in advance which partial design is presumably going to lead to the best design to speed up design search. These heuristics are used to determine the order in which the candidates are tried, so that those producing maximum testability increment and minimum area increment are the first ones to be chosen.

1.2. Search guiding heuristics

The goal of the ordering heuristics is to sort out the candidates for allocation so that the ones leading to the testable design with minimum area are tried first.

As explained before, the algorithm for allocation takes into account the dependence between scheduling and module allocation. So the first module allocation tried is the one provided by the scheduler and only after data paths with non minimum module and area are explored. The same reasoning stands for register allocation:

allocation of minimum register area is tried before exploring the alternatives that lead to data paths with extra registers.

As a result, the heuristics have to follow these module and register allocations as a first choice. After trying the existing alternatives the algorithm will explore the designs with non minimum number of modules and registers that correspond to tradeoffs between module-register area and interconnect area. Besides meeting this constraint, heuristics have to sort allocation alternatives in such a way that the first data paths obtained are the best ones.

Thus the sorting of alternatives is performed in two steps. Application of the first criterion divides the candidates into groups according to the **minimum module and register area criterion**. In the case of register allocation there are only two groups, that is, the one of the alternatives that are part of the partial design and the one of the alternatives that involve adding a new instance to the partial design. On the contrary, application of the first criterion to FU allocation divides each one of the two groups (existing alternatives and new ones) in as many sub-groups as types of FUs can perform the operation of the node. These sub-groups are sorted according to the list of types for the node provided by the scheduler. As a result, the first alternatives tried are the ones in the partial design which correspond to the type of module most suitable for the node; the second ones tried are the alternatives in the partial design which correspond to the second most suitable type; and so on. When all the existing alternatives have been tried, the new ones will be.

After the first sorting, candidates within a group (sub-group) are further classified according to the second criterion, which is composed of a set of optimization rules, namely, **testability maximization rules and area minimization rule** (since minimum module and register area are guaranteed by the first criterion the rule is concerned with interconnect area minimization). The sorting of the candidates for each atomic allocation depends on the set of rules and on the order in which the rules are applied, which is decided for each atomic allocation as explained in [13]. In this paper we assume that testability rules are always applied before the area rule. Testability maximization rules sort candidates by their effect on the controllability and observability of the partial design, as explained next.

2. Estimate of testability

From the algorithm presented in the previous section it is clear that testability estimation is needed **during** allocation. As a result, testability estimation has to be done starting from the information available during allocation, namely *the scheduled control-data flow graph* and *the partial design*. Both are RTL descriptions but the former is behavioral while the later is structural. Thus we use an RTL level metric that was formerly used for behavioral test generation in [11][12]. This metric determines the controllability and observability, which range between 0 and 1, as explained in [13].

Controllability of an input of each element is defined as a function of the minimum distance from any input port to the input of the element. The distance is measured as the number of elements that have to be traversed. Since all the inputs of an element need to be controlled in order to control its output, the distance of the output of an element is one plus the maximum among the distance of its inputs.

Observability of the output of an element is defined as a function of the minimum number of elements traversed from the output of the element to any output port. The distance of the inputs of any element is one plus the distance of the output.

To be used during data path allocation this metric needs some transformation that are explained to the case of FU allocation. When a node has to be allocated to a FU, the elements that store the operands of the node have already been allocated but the elements that will store its results have not. Further, the predecessors of the node, which are scheduled in earlier control steps, have all been allocated and the paths from the input ports to the inputs of the FU can be searched in the partial design. On the contrary, the successors of the node, which are scheduled in future control steps, have not been allocated yet and so the paths from the output of the FU to the output ports have to be searched in the CDFG. So, controllability is defined in the partial design (structure) but observability is defined in the CDFG (behavior).

Some modifications of the metric are required to make a fair comparison of the figures provided by both metrics. In the partial design metric multiplexers do not increment the distance any more since they have no corresponding CDFG element. This modification produces reasonable controllability figures because multiplexers do not decrement controllability in the same amount than FUs and registers do. In the CDFG metric operational nodes increment the distance in two, that is, they are worth double. The reason is that each operational node corresponds to a pair of data path elements: the FU that performs the operation and the register that stores the result. As a consequence the maximum distance used to normalize controllability and observability is also modified. To keep the explanation in this section simple, we assume that normalization is done dividing by 10.

Testability estimations are used in allocation for two different goals. On the one hand estimation of the testability of each data path synthesized is needed to compare it to the testability of other data paths. Then one of the designs synthesized is chosen according to the testability and area estimates. On the other hand estimation of the testability increment of each allocation alternative is needed to sort them and save design space search time.

As a result, there are two testability estimations: the increment of each allocation alternative and the increment in the new partial design due to an atomic allocation. The later use of estimations computes the real testability value of a partial design reached after one of the allocation alternatives has been chosen for an atomic allocation. On the contrary the former use determines the quality of the allocation alternatives so that the ones that will produce the best designs can be selected in the first places. Further details of both estimations are explained next.

2.1. *Testability increment due to atomic allocation*

Every time an alternative is chosen for any atomic allocation a new partial design is produced and the testability increment is computed as the sum of the increment produced for every element in the partial design. Thus it leads to an accurate measure of the testability of the partial design.

A simple CDFG and the partial design obtained after allocation of nodes +1 and +2 are shown in figure 1. Computation of controllability and observability increment is done for the atomic allocation of node +3 to FU +2. This allocation involves adding to multiplexers and two wires (in dotted lines in the figure) to the partial design. Testability increments produce are shown in table 2.

The **controllability increment** due to each atomic allocation is computed as the sum of the increments produced at the inputs of the element that is being allocated and also the controllability increment of the inputs of all the elements that are connected (either directly or through paths) to the output of the allocated element. In the figure the element that is being allocated is the FU +2. There is controllability increment at its inputs because register

R1 is more controllable than the left input (In1) and register R4 is more controllable than the right input (In2). Since both inputs are more controllable the output of the FU +2 increments its controllability. So the elements connected to it have to be explored to find out if there is any further controllability increment. Register R3, the only element connected to the output of FU +2, increments its controllability. So the elements connected to the output of R3 have to be searched. Only FU +2 is connected to the output of R3 and thus there are no further controllability increments.

The **observability increment** due to each atomic allocation is the sum of the increment produced at the output of the element that is being allocated and also the increment of the outputs of all the elements that are connected (either directly or through paths) to the inputs of the former one. In the figure the observability increment is computed taking into account the observability of the CDFG elements in table 1. The notation used is the following: each operational node has two observability figures, the observability of the *output variable* (Vi), which corresponds to the one of the output of the register storing the variable, and the observability of the *node* (node-out), which corresponds to the one of the output of the FU. In the figure the element that is being allocated is the FU +2. There is observability increment at its output because node +3 is more observable than FU +2. Both inputs of FU +2 increment their observability and the elements connected to them, which are registers R1, R3 and R4, are explored. Only the observability of R3 is increased. The elements connected to the input of R3 are explored but no further observability increment is obtained.

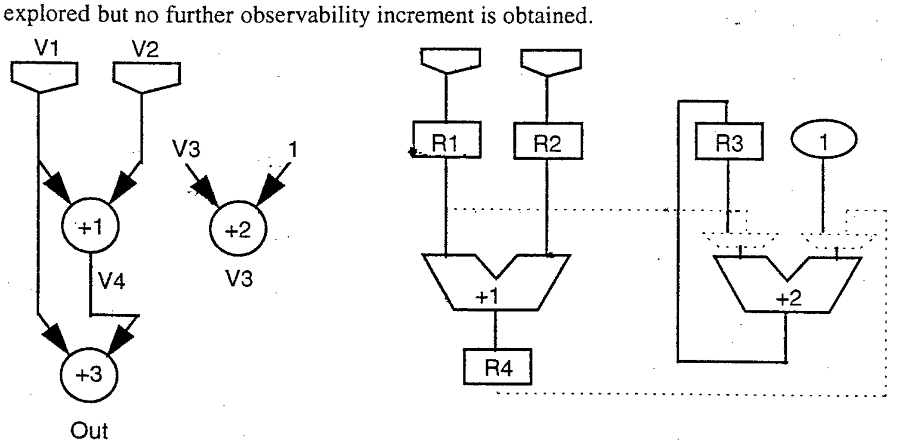


Figure 1: CDFG and partial design for allocation of node +3 to FU +2.

	V1	V2	V3	V4	Out	+1-out	+2-out	+3-out
CDFG Observability	0.8	0.6	0	0.8	1	0.7	0	0.9

Table 1: Observability for the CDFG in figure 1.

	R1	R2	R3	R4	+1In1, +1In2	+2In1, +2In2
co(In) before	1	1	0	0.8	0.9, 0.9	0, 0
co(In) after			0.6			0.9, 0.7
ob(out) before	0.8	0.6	0	0.8	0.7	0
ob(out) after			0.8			0.9

Table 2: Controllability and observability increments in figure 1.

2.2. Testability increment for each alternative

Unlike the testability increment introduced in section 2.1, the testability increment for each allocation alternative is computed **before** selecting an alternative for allocation and producing a new partial design. The testability increment for each alternative is used by the heuristics to sort the alternatives so the ones leading to the best designs are the first chosen. Thus, the testability increment has a different goal than the former estimation: *it has to reflect the effect that each candidate would have on the testability of the complete data path if it was selected.* As long as it accomplish this task it is not important that the increment reflects the present effect that selection of the alternative has on the testability of the partial design. If the alternative is finally chosen, the increment in section 2.1 will be used to compute this present effect on the testability of the partial design.

The second difference between the testability increment due to an atomic allocation and the increment for each allocation alternative is that, for the sake of simplicity, the second one only takes into account the increment on the elements that are directly involved in the atomic allocation, that is, the candidate for allocation and the elements that have to be connected to its inputs.

Taking these considerations into account the controllability and observability increment of each allocation alternative is outlined next. It is also illustrated with an example in figure 2 (FU allocation).

The **controllability increment** of each allocation alternative is computed as the sum of the increment at the inputs of the allocation alternative. There is one exception to this rule when it is applied to allocation of FUs. We do not count the testability increment that comes from connection of a register to an input of a candidate FU if the register belongs to the test paths of the other input or the output of the candidate FU. The reason is that connecting a register to both inputs or to an input and the output does not provide independent test paths for the FU. Thus, this kind of connections is not favored by the heuristics. As an example see the controllability increment for In2 in table 3.

When the element that is being connected to an input is not controllable at all no controllability increment can be obtained in the partial design by selection of any allocation alternative. However, in this case the different allocation alternatives can produce controllability increment in the CDFG if they are controllable. If the variable that increments its controllability is used as input in future allocation steps this increment in the controllability of the variable will result in a more controllable data path. This is the case with the controllability increment for In1 in table 3.

On the other hand, the **observability increment** of each allocation alternative is computed as the increment at the output of the allocation alternative or at the output of the elements that are being connected to it. If the CDFG element is more observable than any allocation alternative, as in the example in table 3, the observability increment is the increment at the output of each alternative. If all allocation alternatives are more observable than the CDFG element the observability increment is the sum of the increment at the output of the elements that are being connected to the alternative. There is one exception to this rule when it is applied to allocation of registers. We do not count the testability increment that comes from connection of a candidate register to the output of a FU if the candidate register belongs to the test paths of any input of the FU. The reason is the same as for controllability increments.

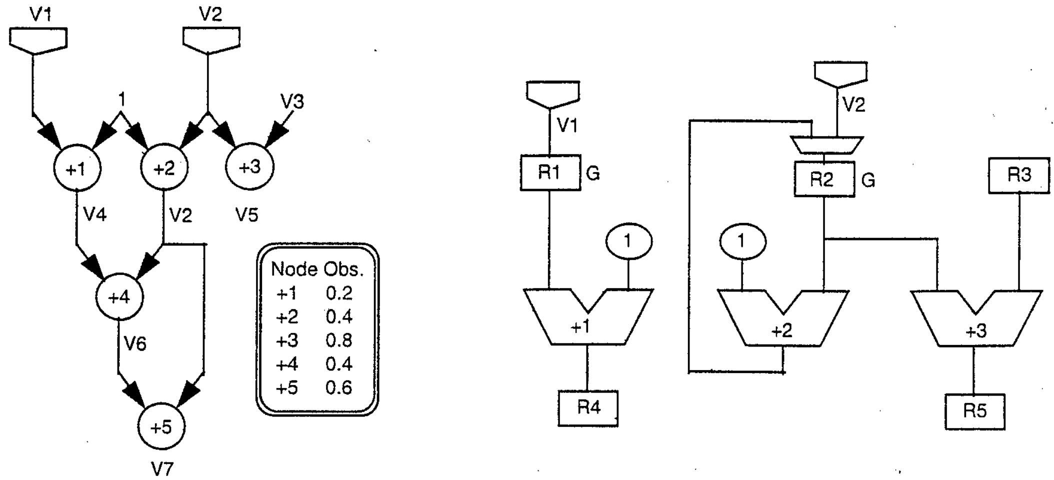


Figure 2: CDFG and partial design for the allocation of node +4.

	Element improved	Increment	FU +1	FU +2	FU +3
Controllability incr. of In1: co(R4) = 0	Variable V6	Controllability of candidate-In1	0.9	0	0.9
Controllability incr. of In2: co(R2) = 0.9	Input of candidate	co(R2) - co(candidate-In2)	0.9	0	0 (R2 in the test path of In1)
Observability incr: ob(+4) = 0.4	Output of candidate	ob(+4) - ob(candidate-out)	0.2	0	0

Table 3: Testability increments of the alternatives for allocation of node +4 in figure 2.

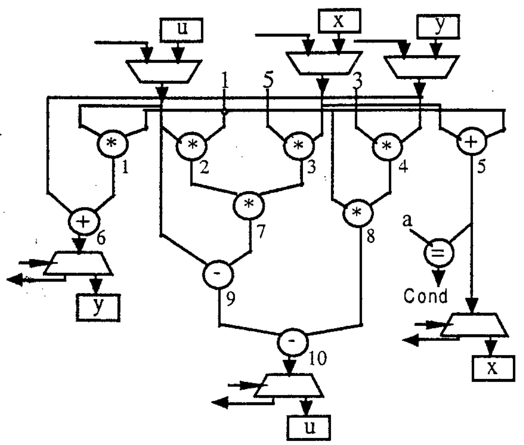
3. Results and conclusions

The behavior of the well known HLS benchmark Differential Equation solver [14] is used to show how our algorithms work. The CDFG, the scheduling and tentative module-register allocation provided by the scheduler and the observability of the output of the operational nodes are included in figure 3. The controllability of all of them is very low due to the existence of many constants.

The behavior has 3 input ports (x, y, u) and 3 output ones (x, y, u). There are also 10 operational nodes, each one executed in 1 control step, that perform 6 multiplications, 2 additions and 2 subtractions. The minimum area set of modules is made of 2 multipliers, that take up almost 80% of the total circuit area, a ripple-carry adder and a ripple-carry substracter.

The behavior includes 2 data dependencies that make registers storing x and y be self-adjacent. Besides, variables x, y, u generated by nodes 5, 6, 10 respectively have to be allocated to the same register than before. Thus register allocation for the output of those nodes is fixed.

In spite of all these constraints our allocation algorithms succeed in obtaining some self-testable data path without CBILBO registers and small interconnect area. Before presenting the data paths synthesized by our algorithms we will analyze the different data paths that can be obtained to evaluate how our heuristics work. In figure 4 the two possible FU allocations are presented for the addition and substracion nodes. In figure 5 the two best FU allocations for multiplication nodes are included.



C.Step	Node
2	2, 3, 5
3	4, 7
4	1, 8, 9
5	6, 10

Modules: 2*, +, -
Registers: 6

Node	Obs.
1	0.7
2	0.3
3	0.3
4	0.5
5	0.9
6	0.9
7	0.5
8	0.7
9	0.7
10	0.9

Figure 3: CDFG of the Differential Equation with scheduling inf.

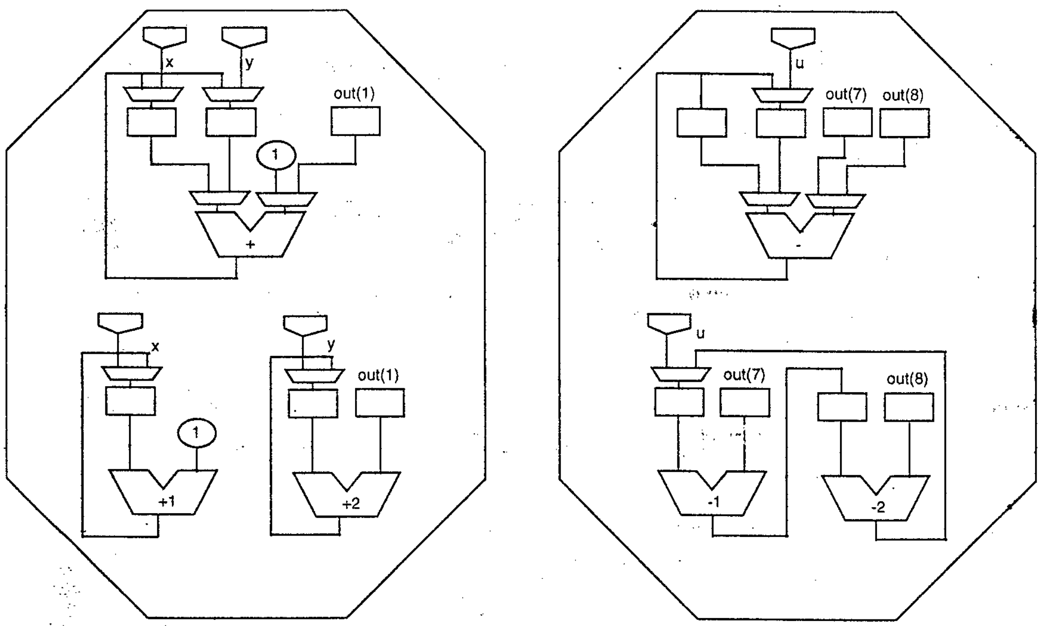


Figure 4: Allocation alternatives for additions and substractions.

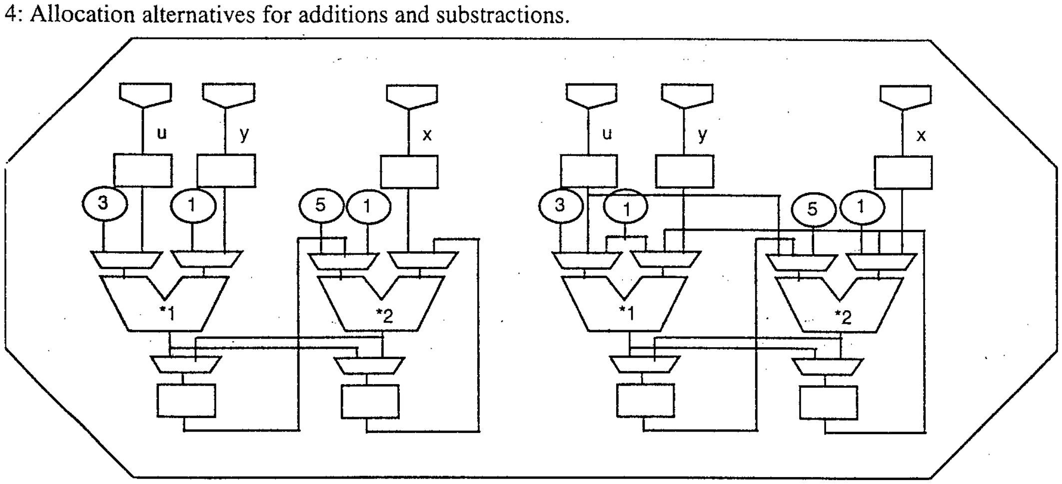


Figure 5: Best allocations alternatives for multiplications.

As shown in figure 4, allocation of adders for nodes 5 and 6 is quite constraint. The inputs to node 5 are the input variable x and constant 1, and its output is the output variable x. So register allocation is fixed for this node. The inputs to node 6 are the input variable y and the output of node 1 and its output is the output variable y. Module allocation for these nodes can choose between having one or two FUs. The former choice, the first one due to the greedy nature of the first sorting criterion, is more testable than the later one, which needs two CBILBO registers in order to test the modules. In the data paths obtained by our algorithm we observe that the choice of having two adders is never included in them, since CBILBO registers degrade circuit performance and are very costly.

On the contrary, allocation of substracters for nodes 9 and 10 produces two acceptable alternatives that are found in the data paths synthesized by our algorithms. The first choice consists on having only one FU to perform both operations and it implies having 2 self-adjacent registers, which do not need to be implemented as CBILBO. The second choice consist of using two modules, one for each node, so the test time and test area are locally increased (the increment depends on the rest of the data path). The increment on module area (a simple ripple-carry substracter) is balanced with the decrement on interconnect area (two 2-to-1-multiplexers).

Finally, in figure 5 we only include the two alternatives that can appear in the best designs, because they correspond to the best value of a quality parameter (area and testability). The remaining options are bigger and less testable than the ones in the figure and thus they cannot appear in the best designs.

The left option corresponds to the smallest one that can be tested while the right drawing is slightly bigger but also has smaller test time.

Once the alternatives have been analyzed, we will introduce the data paths that our algorithm found for this CDFG. The order in which they were obtained is also important, so we use it to label them.

Table 5 includes the number of modules, multiplexers, wires and the information needed to characterize testability (BIST registers and test sessions). Designs 1 and 2 correspond to the more testable option for multiplications and the two options for substractions. The ther ones correspond to the smaller choice for multiplications. All can be tested without CBILBO registers in spite of the data dependencies in the behavior and the constraint register allocation. Note that the first solutions explored are the best ones but among them the more testable ones are explored before the smallest ones. This is due to the fixed order for rules application.

# Design	# FUs	# Mult. & wires	BIST registers	# Test sessions
1	2*, 1+, 1-	4 4-to-1, 9 2-to-1, 34	3 G, 2 S, 1 B	3
2	2*, 1+, 2-	4 4-to-1, 7 2-to-1, 34	2 G, 1 S, 3 B	4
3	2*, 1+, 1-	1 4-to-1, 11 2-to-1, 30	3 G, 1 S, 2 B	3
4	2*, 1+, 2-	1 4-to-1, 9 2-to-1, 30	2 G, 4 B	3

Table 5: Data paths obtained for the CDFG in figure 3.

In table 6 we also include the testability estimation of the first solution. We observe that in spite of the low testability of the behavior, the data path is testable due to the allocation performed.

	R1	R2	R3	R4	R5	R6	*1	*2	+	-
Controllability	1	1	1	0.8	0.8	0.6	0.9, 0.9	0.9, 0.9	0.9, 0.7	0.9, 0.7

Observability	1	1	1	0.8	0.8	0.8	0.7	0.7	0.9	0.9

Table 6: Controllability and observability of design 1 in table 5.

References

[1] M. Abramovici, M.A. Breuer and A.D. Friedman, "*Digital System Testing and Testable Design*", Computer Science Press, 1990.

[2] P.H. Bardell, W.H. McAnney and J. Savir, "*Built-In Test for VLSI: Pseudorandom Techniques*", John Wiley & Sons, 1987.

[3] B. Koenemann, J. Mucha and G. Zwiehoff, "Built-In Logic Block Observation Techniques", *Proc. of the ITC*, pp. 37-41, October 1979.

[5] L.T. Wang, E.J. McCluskey, "Concurrent Built-In Logic Block Observer (CBILBO)", *Proc. of the Int. Symposium on Circuits and Systems*, pp. 1054-1057, 1986.

[6] D.D. Gasjki, N.D. Dutt, A.C-H. Wu and S.Y-L. Lin, *High-Level Synthesis Introduction to Chip and System Design*, Kluwer Academic Publ., Boston, 1992.

[7] R. Moreno, R. Hermida, M. Fernández and H. Mecha, "A unified approach for scheduling and allocation", *INTEGRATION, the VLSI journal*, vol. 22, 1997 (in press).

[8] J. Septién, D. Mozos, J.F. Tirado, R. Hermida, M. Fernández and H. Mecha, "FIDIAS: An integral approach to high-level synthesis", *IEE Proc. on Circuits, Devices and Systems*, vol. 142, pp. 227-235, August 1995.

[9] H. Mecha, M. Fernández, J.F. Tirado, J. Septién, D. Mozos and K. Olcoz, "A Method for Area Estimation of Data-Paths in High Level Synthesis", *IEEE Trans. on CAD*, vol. 15, pp. 258-265, February 1996.

[10] K. Olcoz, J.F. Tirado, D. Mozos, J. Septién and R. Moreno, "Data Path Structures and Heuristics for Testable Allocation in High Level Synthesis", *Microprocessing and Microprogramming*, vol. 39, pp. 263-266, 1993.

[11] J.F. Santucci, G. Dray, N. Giambiasi, M. Boumedine, "Methodology to reduce computational cost of behavioral test pattern generation using testability measures", *Proc. of the 29th IEEE/ACM DAC*, pp.267-272, June 1992.

[12] J.F. Santucci, G. Dray, M. Boumedine, N. Giambiasi, "Methods to Measure and to Enhance the Testability of Behavioral Descriptions of Digital Circuits", *Proc. of the 1st IEEE Asian Test Symposium*, pp.118-123, 1992.

[13] K. Olcoz, J.F. Santucci, J.F. Tirado, "Integration of behavioral testability metrics in high level synthesis", Embedded System Applications, pp. 73-87, Kluwer Academic Publ., 1997.

[14] P.G. Paulin, J.P. Knight and E.F. Girczyc, "HAL: A Multi-Paradigm Approach to Automatic Data Path Synthesis", *Proc. of the 24rd ACM/IEEE DAC*, pp. 263-270, June 1986.

About Solving Complex Problems

Thierry CARLOTTI , Mostafa HATIMI
Université de Corse
SDEM URA - CNRS 2053
Quartier GROSSETTI
20250 CORTE

Phone: (33) 04.95.45.01.09 — Fax: (33) 04.95.61.05.51
email:{carlotti, hatimi}@lotus.univ-corse.fr

The goal is to give at user an environnent being able to describe complex problems. Then, we try to solve these problems. To lead the resolution, we decompose basis complex problem by canonical sub-problems. For that, we dispose an existant sub-problems library's, each problem being attached with its solve method.

Once our complex problem as been decomposed, each sub-problem is typed and attached with its representative library member. At this level, classify a problem is equivalent to arrange it at its place. A complex problem is represented by its complexities, variables of states, conditions bounds links to the environnement in which it evolves; behaviour in time.

It is necessary to take into account these three main aspects: morphological (constitution), spacial (number of variables) and temporal (resolution time).

In this paper, we present a reflection for the tool to conceive. This reflexion is structured with three parts. Each part, we go up main points on which we will lean to permit an implementation of a such tool that will work intended to help user face to this type of problem. This paper is organized as follows; first, we expose briefly one state of art ; second, we present tools and classification method ; thirth we describe our approach towards a classification model and resolution.

1 State of art

Tool to be develop is composed of some several blocks that collaborate each to the a same achievement that is the final task, provide an answer's complex problem response to the user.

1.1 Coding interface

Software interface allows to code the problem, that's happens once the validated foreclosure. A block specialized with coding will be charged to transcribe the statement towards the next block described below. Differents strategies for information coding are frequently presented, in most cases it is used a binary coding alphabet on initial information [Dumeur 94]. Coding brings back to form a chain of characters based on binary characters or use of others coding methods by using relational-schemata [Collard 96].

1.2 Process of ranging

Once the problem is coded, then it must be classified and "typed". This function is achieved with the classifier block, which allows to arrange the problem with a representative library's member. A classifier can be considered like a deduction system able to recognize, that means he can distinguish entities [Ducourneau 96].

1.3 Library's members evaluation

Before being classified, the representative library's member is appraised [Escazut 96]. This evaluation is necessary to allow classification of the problem, if the representative member is considered like bad, the classement will not succeed, in such situation problem can very well to [Dorme 96]:

- be approach, in this case it is attached to a representative library's member who is less representive,
- be classify by the creation of a new representative library's member.

Representative's member estimation is achieve by an evaluation principle. If the problem is approach, it arrives that several representative's member are considered as good to take possession of the problem. With such cases some authors introduce a noise function with the evaluation function. It avoid to have library's representative members having the same evaluation [Dumeur 94].

With the case where a new representative member will be create, the size of the library will increase. Well, this fact is unavoidable, we try to limit at maximum such type of situation, our wish being to keep a library composed with a constant representative library's members. Thus, the problem will be more often approach by several representative library's members if they are considered as good.

2 Classification method and tools

Purpose is to show why we have choosen some tools rather that others available in the litterature.

2.1 Why not neurals networks?

Neural networks principles allows participation of several neurons for the achievement of a common stain. Each neuron is triggered by a activation threshold. Generally, a base of learn allows to fix weights on neuron inputs, beside activation of each neuron depends of inputs weights values [Jodouin 93].

At our level, difficulty is that we should work in a non-supervised mode, we cannot uses data that should use as experiences basis to permit the neural network training. More, achieve a classement with a non-supervised mode sometimes provokes errors when singular situations occurs if Hamming distance between two representative library's member is lower than the one specified by the user [Amy 96]. Also, essential point is that we wish to transport variables instead of values. It is therefore impossible, with these considerations to be able to use neural network in our classification block

2.2 Why not genetic algorithms?

Genetic algorithms use chromosomes composed with genes. Thus, they permit to transport a gene sequence; this sequence can represent for us our codification problem. Also, others tools permit new chromosomes generation from a population; we could use these genetic opertors specificity to permit the creation of a new representative library's member for a problem that doesn't have been classified while using a traditional method [Escazut 96]. So we will us it.

3 Toward a classification model

We present here the different blocks that participle to the realization of our objective, we are going to present a model putting in work "classification" and "evaluation".

We will retail, first the simple problem's classification mechanism to one representative member of our library. In section 3.2, when we talk about of "problem" unterstand simple problem.

3.1 Attribute member aspects

Each member of our library is composed with attributes. There are N attributes of each member. We call $f_k(o)$, evaluation function of k attribute for an o object. With that $f_k(o)$ expression is shown at (1)

$$f_k(o) = w_k(o) \cdot Attr_k(o) \tag{1}$$

where:

- $w_k(o)$ is the weight attribute for a o library's member, here value is fix to a same integer coefficient,
- $Attr_k(o)$ is the k attribute value for a o library's member, here value is binary,
- $f_k(o)$ evaluation function for k attribute of a o library member.

Thus, total evaluation function that we call $F(o)$ is shown at (2):

$$F(o) = \sum_{j=1}^{N} w_j(o) Attr_j(o) \tag{2}$$

- where $w_j(o)$ are weights attributes for an o library's member, here values are fixed to a same integer coefficient,
- where $Attr_j(o)$ are attributes values for a o library's member, here values are binary ,
- where N is the number of o attributes member.

3.2 Searching for a good representative library's member

Purpose is to find a representative member of our library that will represent the problem, or one that resembles to him enought strongly, in the opposite case it is about to create one from the best representative members evaluate like making part of our library. The $D(o)$ expression shown at (3) is used for the evaluation, and is introduce like a distance from the problem to the representative's member library.

$$D(o) = |\sum_{j=1}^{N} w_j(o) Attr_j(o) - w_j(Ci) Attr_j(Ci)| \tag{3}$$

- where $w_j(o)$ are weights attributes for our problem o to be class, here values are fixed to a same integer coefficient,
- where $Attr_j(o)$ are attributes values for our problem o to be class , here values are binary values,
- where N is the number of o attributes problem,
- where $w_j(Ci)$ are weight attributes for a Ci library's member, here values are fixed to a same integer coefficient,
- where $Attr_j(Ci)$ are attributes values for a Ci library's member, here values are binary values,
- where N is the number of o attributes member.

Three cases can appear :

- case where one representative member has a evaluation distance that is $D(o) = 0$ that means, we have find the ideal representative library's member for our problem,
- case where no representative member are evaluated as a member having ability to represent the problem, therefore a selection of the best components evaluated as such in the library is operated. If evaluation distance $D(o)$ is below than a certain threshold $S0$ to fix, then, the best representative member is elected like being the best able to represent the problem, thus problem is attached to him.
- case where best representative members have a evaluation distance $D(o)$ that is over the $S0$ threshold, we select the best representative members on their evaluation values, and we applies on them genetic operators, with that we generate a new generation members, then will evaluate them.

At the end of the evaluation, if it is found a new representative member having a evaluation distance $D(o)$ below $S0$ threshold, we add it in our library. With an opposite case, we proceed again by applies genetic operators on the best representative members. Once the representative member is added at the library, we can say that it is the new representative member that represents the problem.

To avoid each time to use the same representative members for the generation process, in the case which previous generation hasn't been capable to produce a good representative member, a noise function is apply on the evaluation function, purpose is to make fluctuate this evaluation value around a average value. By this way, we will not take same representative members for two successive generations

3.3 Knowledgment of more used library's members

Purpose is the knowledgment of the more used representative members for accelerate our global gait, instead to proceed by evaluate all the members of our library, we will start in priority with those that are more frequented. By this way, we will have more chance to find the best member of the library, or perhaps the ideal for our problem.

Another interesting aspect is to be able to detect members of our library rarely used , or not at all, by this manner we'll able to replace them by others when a new member is added at our library.

To allow the knowledgment of these representative members, we use a Bricket Brigade algorithm variant[Holland 75], by fixing identical strengths at each member of our library at the beginning. When a new representative member is added at our library, its strenght will be averaged values fathers's stenghts.

4 Conclusion

The goal fixed along this study is to give some results to solve complex problems, we develop a method based on the classification existing results. We plan to complete this work with an efficient coding technic to improve the structure implementation of data with an resolution's optimal algorithm.

References

[Dumeur 94] : R.Dumeur "Synthèse de Comportements des Animaux Individuels et Collectifs par Algorithmes génétiques". Departement informatique - Institut d'intelligence Artificielle. Universite de PARIS 8.

[Collard 96] : P.Collard 'Relational shemata : "A way to Improve the Expressiveness of Classifier" Laboratory I3S CNRS-UNSA 250 Av. A. Einstein, Sophia Antipolis, 06560 Valbonne, France.

[Ducourneau 96] : R. Ducourneau "Des Langages à Objets aux Logiques Terminologiques : les systèmes classificatoires". LIRMM Unité Mixte CNRS - Université de Montpellier II, 161, Rue Ada.

[Escazut 96] : C.Escazut "Typology of Boolean Functions using Walsh Analysis" Laboratory I3S-CNRS UNSA 250 Av. A. Einstein, Sophia Antipolis, 06560 Valbonne, France.

[Dorme 96] : R.Dorme "Nouveaux Opétateurs Génétiques Appliqués à SAT" EERIE, Parc Scientifique Georges Besse F-3000 Nimes.

[Jodouin 93] : J.F Jodouin "Reseaux de neurones et traitement du langage naturel". Université de Paris-Sud. Orsay.

[Amy 96] : 'B. AMY "Recherche et Perspectives Dans le Domaine des Réseaux Connexionnistes" Laboratoire LEIBNIZ-IMAG - 46, avenue F.Viallet.

[Holland 75] : "Adaptation in Natural and Artificial Systems" University of Michigan.

Neural networks for the study of natural systems

Ciccoli François-Xavier, Delhom Marielle, Chiari Frédéric, Santucci Jean-François
University of Corsica
SDEM-Systèmes Dynamiques Energétiques et Mécaniques
URA-CNRS 2053
BP 52-Quartier Grosseti-20250 Corti-France
E-mail : {ciccoli, delhom, chiari, santucci} @univ-corse.fr

Introduction

This paper presents an original approach based on neural networks to study natural systems.

Generally the study of this kind of systems is performed using determinist methods. For example, at the University of Corsica [DEL 1995; DEL 1997], we have developed an object oriented modelling and simulation approach in order to analyse the hydrologic behaviour of a catchment basin. In our approach, the system is modelled in several sub-systems, called basic components, which have their own behaviours. We have also defined a simulator architecture based on discrete event and object oriented simulation.

Besides, a simulation software has been realised. Particularly, it allows to process independently the modelling and the simulation parts. So, this makes easier the definition and the modification of the model. Owing to this explicit separation between the modelling and the simulation aspects, several models can be implemented and studied without any modification of the simulation software.

Nevertheless, this approach presents some limits :

- this kind of modelling needs to know with accuracy the physical behaviour of each sub-systems and in the case of the study of natural systems, these behaviours are not always available ;
- the modelling of natural systems requires to define several parameters which are fit for this system but if one wants to adapt the realised model to another system, he has to re-define these parameters ;
- finally, it's difficult to take into account all the input data.

Faced with these limits we propose to integrate neural networks in order to complete our approach with the help of neural networks. Indeed, the connectionist theory present these interesting advantages :

- if input and output data are available, it's possible using a learning process to predict the behaviours of systems and then analyse most complex systems ;
- the model realised using this approach is adaptable and can be reused to study another system ;
- it's possible to take easily into account all the input data.

The paper is organised as follows:

The first section presents some general notions concerning neural networks and a classification.

In section 2, we present a class of networks, called supervised learning network, particularly adapted to the study of natural systems.

We finally conclude, in section 3, by briefly presenting the current status of our works, future investigations are also discussed.

1. Basic notions

1.1 Presentation of neural networks

The first work concerning neural networks was inspired by a brain modelling which is nowadays the most efficient known system for data analysis. Also, a network is composed by units called formal neurons which are connected using weighted links. A learning process is implemented in order to simulate the human reasoning.

As the propagation of nerve impulse, the activation progress throughout the network. Every activated unit gives its rate and a global activation emerges from the network.

The following graph presents a classic type of network called Multy Layer Perceptron [ROS 1962; JOD 1994; CAR 1992].

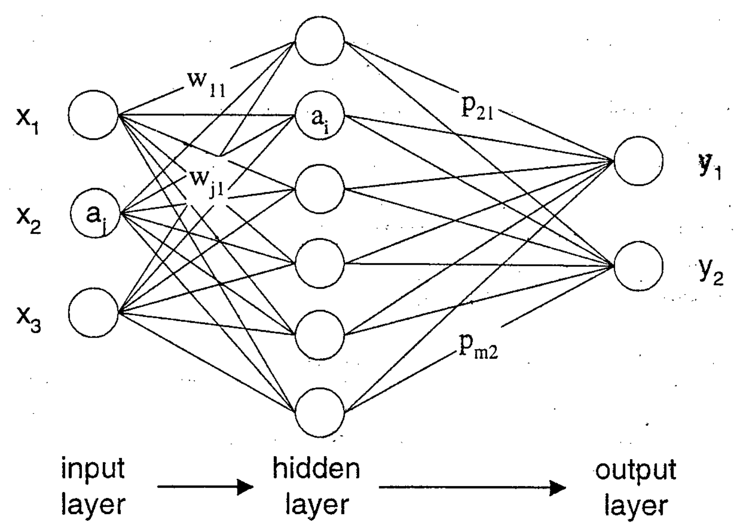


Figure 1 : Organisation of a neural network

For example, the activation a_i of the neuron i is controlled by a stochastic rule: the sigmoidal function $s(x) = 1/[1+\exp(-\alpha x)]$. The activation of a given neuron may be a weighted sum of the activation of the linked neurons, i.e. $a_i = \sum_j w_{ji} a_j$.

1.2 Classification of networks

As the connectionism science covers many areas (biology, classification, recognition, environmental science, ...) it is not possible to realise a standard classification. However, to be in adequacy with the main themes of researches, we have chosen to present a neural networks classification according to their training method [AMY 1996].

It is possible to distinguish three main types of networks:
· the not supervised networks;
· the supervised networks;
· the reinforced learning networks.

The first training rule proposed by Hebb in 1949 [HEB 1949], is not supervised and was constructed to avoid a random choice of network parameters. However, these rules just order in different categories input vectors proposed to the network. This classification is realised according to input structure vectors and not to their qualities. Nevertheless, with encoding information techniques, such networks have proved their capacities to compete with best classifiers [JOD 1993].

The supervised training allows to extend the application domain of the connectionism. It consists in teaching to the network a reference behaviour. That is to say that one gives the expected answer and using an error minimisation it adjusts its parameters. The convergence is obtained using a gradient descent method, the famous backpropagation of gradient, this one is now replaced by a faster method of second order [SZI 1995; SZI 1997].

One also finds a mixed method called reinforcement learning developed to study applications having several possible solutions. These algorithms allow to estimate the quality of the answer in order to direct the network toward the best solution.

2. Supervised Learning Network

The natural systems survey often needs to approximate a not-linear function, sometimes discontinuous. In this setting, we are more especially interested by supervised training networks and more specially :
· the Radial Basis Functions;
· the Recurrent Cascade Correlation.

2.1 Radial Basis Functions

This type of network is dedicated to the approximation of linear parametric functions, semi-linear or not-linear. Function f(x) is approached by a combination of elementary not-linear functions.[AMY 1996]

$f(x) = \sum_i w_i \varphi_i$ where φ_i are basis functions generally gaussian

w_i are calculated weights.

The approximation of a continuous function allows to justify theoretically the network structure. Unlike, there is no established theory concerning the approximation of discontinuous functions, fortiori not-linear, which are the more often used to model natural systems. In this case, the structure of the neural network is defined according to experimental results [LAB 1993].

2.2 Recurrence Cascade Correlation

This kind of network is also called dynamic or incremental. Indeed, the addition of neurons starting from a minimal configuration is automatically done according to the complexity of input/output couples. Moreover, this networks is recurrent, indeed, the outputs of the hidden layer at time t are fed back for use as additional network input at time t+1. These time delay networks are very interesting for study of natural systems.

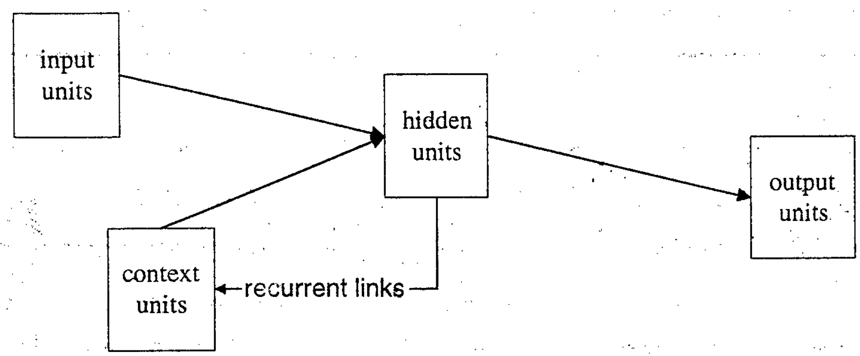


Figure 2 : Structure of a recurrent network

3.Conclusion

We present in this paper an original approach based on neural network to study natural systems. We have already proposed a determinist approach based on oriented objet modellisation and simulation, but in the case of the study of such systems, this approach present some limits.

In order to overcome these limits, we propose to use neural networks and more precisely supervised learning networks.

Today we are applying the different notions presented in this paper to the study of the hydrologic behaviour of a catchment basin. Particularly, we want to increase the "reuse" of the model using neural networks in order to adapt the system's parameters. We intend in a near future to integrate notions presented before in our simulator architecture.

Bibliography

AMY B. (1996), "Recherches et perspectives dans le domaine des réseaux connexionistes", Rapport interne du laboratoire Leibniz-Imag.

CARLING A. (1992), Introducing neural networks, Sigma Press, New York.

DAVALO E. – NAÏM P (1993), Des réseaux de neurones, Eyrolles.

DELHOM M. - BISGAMBIGLIA P.A. - SANTUCCI J.F. (1995), "Modeling and Simulation of Discrete Event Systems : the study of the hydrologic behavior of a catchment bassin". IEEE international conference on Systems, Man and Cybernetics, Vancouver, Canada, Octobre 22-25, Vol. 5, p. 4191-4195.

DELHOM M. - BISGAMBIGLIA P.A. - SANTUCCI J.F. - AIELLO A. (1997), "Modélisation et Simulation Orientées Objet - application à l'étude du comportement hydrologique d'un bassin versant". Revue l'Objet, Editions Hermès, Numéro I, Volume III, p 65-.83.

HEBB D. (1949), The organisation of behavior, Wiley.

JODOUIN J.F. (1993), Les réseaux de neurones, Hermes.

JODOUIN J.F. (1994), Les réseaux neuromimétiques, Hermes.

LABBI A., (1993) "Sur l'approximation et les systèmes dynamiques dans les réseaux neuronaux" Thesis, INPG, Grenoble, France.

ROSENBLATT F. (1962), Principles of neurodynamics, Academic Press, New York.

SZILAS N. (1995), "Apprentissage dans les réseaux récurrents pour la modélisation mécanique et étude de leurs intéractions avec l'environement", Thesis, INPG, Grenoble, France.

SZILAS N. (1997), "Les réseaux récurrents supervisés: une revue critique". Rapport interne, INPG.

GENERATION OF VALIDATION TESTS FROM BEHAVIORAL DESCRIPTION IN VHDL

Ewa Idzikowska
Technical University of Poznan; Dpt. of Control, Robotics and Computer Science; Poland
email: eidz@agat.sk-kari.put.poznan.pl

For the modeling of logical systems on the high behavioral level, hardware description languages are used. The model is a program that represents the algorithmic behavior of the system. It must include a provision for concurrency of execution, because logic signals flow in parallel. This requirement in VHDL is handled by the process construction. The behavior of the design is proved, by running simulations using test patterns. This tests selection is generally done manually by the designer and is thus often incomplete and inadequate. The method proposed in this paper is a contribution to the automation of this process.

Introduction

Functional testing aims at testing the behavior of the circuit under test. Because behavioral models are available early in the design cycle, the generated test sets can be applied for design validation. Validation of the design in early phases reduces the risk of undetected design errors.

Behavioral description of logical systems can be divided into two categories: procedural and non-procedural. In a procedural description, all the statements are executed sequentially as in a conventional program. In a non-procedural description the behavior is represented by a set of process statements. These processes can be executed in parallel when an event appears on their sensitive variables.

The VHDL behavioral model describes the logical system as a set of such processes. In this way concurrent activities of the system are modeled.

Usually, two behavioral approaches to test generation from VHDL have been explored. In the model perturbation approach, high-level fault models are first determined ([1], [3]). Next, a test generation algorithm, which finds tests for these faults is defined. A second approach to test generation uses I/O model ([2], [4]). Here one does not use a fault model, but merely tries to exercise all input/output paths through the model.

The test generation method presented in this work searches for data paths from circuit inputs to outputs, examines the related control inputs and input sequences to drive those transfers and exercises the circuit with characteristic data.

The VHDL model represents two aspects of the process: computation and control. Computation represented by data flow part of the model can be modeled by data flow graph. Control represented by control flow part of the model dictates the partial ordering of data operations and can be modeled by control flow graph. Both models are used for test generation.

The process of the test generation consists of two steps: fist - circuit behavior selection and second - test pattern selection.

Control flow and data flow

Control flow and data flow are represented by graphs, derived from source code.

Def. 1 Data flow graph [4]

The data flow graph DF is a directed graph, not necessarily strongly connected, whose nodes correspond to signals, variables or constant values. Nodes drawn as rectangles are terminals (primary input or output signals), nodes drawn as circles refer to internal variables. Directed edges of the DG show the flow of data from node to node and are labeled with numbers of assignment statements that effect transformation and flow of data.

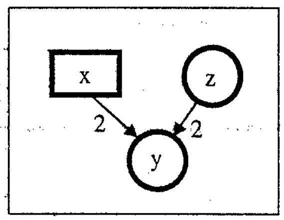


Fig. 1. Data flow graph.

For example a VHDL statement y:=z+x (x - input signal, y and z - internal variables) labeled with number 2 is represented by the following graph (Fig. 1.).

Control is represented by a control flow graph. This model expresses the different phases involved in the execution of the processes and is created using Petri nets.

Def. 2 Control flow graph

The Petri-net representation of the process is a digraph CG derived from the VHDL source code, with mapping of VHDL code statements to the places and actions to the transitions:
CG = (P, T, F, M_o, D, C)
where:

P	$= \{ p_1, p_2, \ldots, p_n \}$	- a finite set of places,
T	$= \{ t_1, t_2, \ldots, t_m \}$	- a finite set of transitions,
M_o	$= \{ m_1, m_2, \ldots, m_n \}$	- an initial marking,
D	$= \{ d_1, d_2, \ldots, d_m \}$	- a finite set of time intervals associated with transitions,
C	$= \{ c_1, c_2, \ldots, c_m \}$	- a finite set of conditions associated with transitions,
F		- a control flow relation.

When a control place p_i holds a token, a control signal will be sent to control the corresponding arcs in the data flow graph. As there could be more than one control place that holds tokens, there exist multiple control signals that control parallel operations in the data graph. The temporal relation between the control signals is defined by a partial ordering structure, which is captured by the control flow relation F. Normal Petri net models do not contain an explicit notion of execution time, hence it is impossible to determine the signal flowing times. To solve this problem, we have introduced timing into the control flow model. A transition will be fired when it is enabled, after some period of time (d_i).

To represent the situation where the control flow is influenced by the results of some data graph operations (a conditional branch, for example) we must be able to use a condition signals to direct the control flow. To solve this problem, transitions may be guarded by conditions (c_i) produced from the data flow. The conditional transition will be fired immediately when it is enabled (normal Petri net) and its condition is true.

Control flow graph of a single process

In digital systems a logical process will frequently pause in its execution while waiting for an event to occur or a time period to elapse. It means that each process could be either in activity state or in wait state (suspended). The form of wait construct in VHDL [1] is:

WAIT ON signal changes UNTIL condition clause FOR time out

The statement suspends the process until a signal in the sensitivity list changes, (e.g. the value of the sensitive variable is not stable – different at times *t* and *t-1*), at which time the condition clause is evaluated. The condition clause is an expression of type Boolean and if it is TRUE, the process is resumed. The time out clause sets the maximum wait time after which the process will resume. A form *WAIT ON C* is equivalent to the form *PROCESS(C)*.

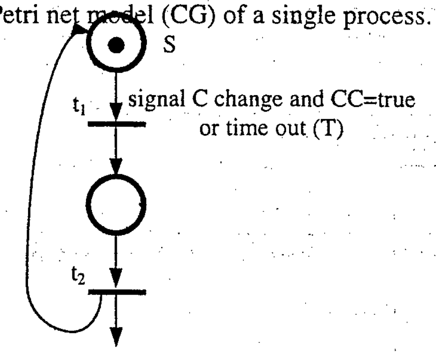
The Petri net model of a single process is shown in Fig. 2. Each process has a place S, which represents a state of the process. A token located in this place denotes the suspended process. The place without a token represents the active process.

A transition t_1 is an input transition to the process. The following events can be associated with this transition:

- signal changes,
- condition clause,
- time out.

A transition t_2 is the process output transition. After firing t_2 the process is suspended i.e. a token is in the place S.

Fig. 2. Petri net model (CG) of a single process.



Transition t_1 is fired when the value of the signal C at the time *t* is different from the value of C at the time *t-1* and when the condition clause CC is true, or regardless of the conditions after a delay T.

The control flow model of several processes is obtained by interconnecting the control flow models of each process.

Test generation

The process of the test generation consists of two steps: first - test case selection (behavior selection) and second - test pattern selection.

During a test generation process we want to derive complete validation tests. The goal for a complete test is to exercise the model in such a way that every possible CG path is accessed with all possible data and the results have to be observable at an output. For the selection of a complete set of test cases we must first cover each path of data flow graph from input to output at least once, and then translate these paths into corresponding list of CG nodes.

For single-process models we can define the CG paths from the control flow graph input nodes through the nodes selected in data paths, into the output nodes of the process using Petri nets mechanism. The VHDL statements represented by the nodes of this path have to be executed in the given order to meet the intended behavior. The test case consists of this nodes sequence.

For a multi-process model such paths can go not only through one process, so we can determine the list of processes to be crossed in order to reach a primary output. Thanks to a Petri nets formalism we can also determine the parallel execution of the active processes.

As an example consider a VHDL model from Fig. 3. This model consists of two processes: process A and process B.

```
entity D_AND is
generic(DEL1, DEL2: time);
port(NCLR, D, CLK: in bit; OU: out bit);
end D_AND;
architecture Data_flow of d_and is
signal Q: bit;
begin
A:process(NCLR, CLk)
  begin
   if NCLR='0' then                            A1
      Q<='0'after DEL1; else                   A2
      if (CLK = '1' and not clk'stable) then   A3
         Q <= D after DEL1;                    A4
      end if;
   end if;
  end process A;
B:process(Q,D)
  begin
   OU <= Q and D after DEL2;                   B1
  end process B;
end data_flow;
```

Fig. 3. VHDL source code

The data flow graph of the VHDL model from Fig. 3. is shown in Fig. 4. This graph is used in data path oriented test case selection. There are three possible data paths from primary inputs to primary output:

1. A2-B1,
2. A4-B1,
3. B1.

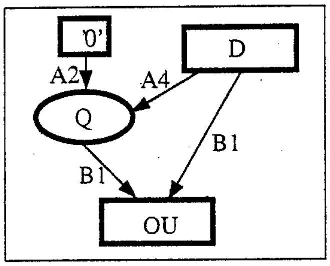


Fig. 4. Data flow graph.

The path *A4-B1* shows, that in order to drive data from input D to output OU statements represented by control flow graph nodes *A4* and *B1* have to be executed. Thus, the test case consists of this nodes sequence.

The control flow graph is shown in Fig. 5. It is composed of control flow graphs of two processes: A and B.

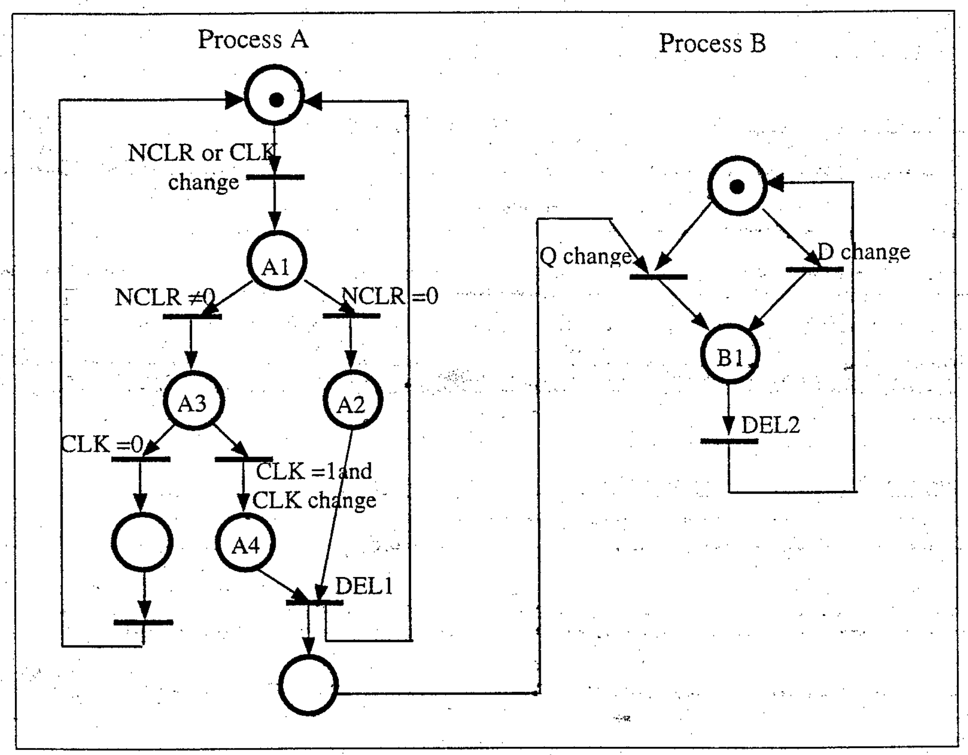


Fig. 5. Control flow graph.

The nodes A4 and B1 (the source code statements Q <= D after DEL1 and OU <= Q and D after DEL2 respectively) will be executed, if the following conditions will be satisfied:

- Not NCLR'stable or not CLK'stable,
- NCLR≠0,
- CLK=1 and not CLK'stable,
- not Q'stable or not D'stable .

time	NCLR	CLK	D	Q	OU
t_0	x	0	1	Q	Q
t_1	1	1	0	1	0

Fig. 6. Test pattern for the data path A4-B1.

The test pattern for the path A4-B1of the data flow graph is shown in Fig. 6.

Summary

In this paper an approach for behavioral test pattern generation based on a formal, non-procedural VHDL description is presented.
Two models have been defined: data flow model and control flow model. The first one describes the links between the inputs and outputs of all the processes. The second one represents the conditions under which the processes are active. The control flow dictates the partial ordering of data path operations. Next we developed principles, which allow to generate test sequences. At the present time we are still working on the implementation of this method.

References

[1] Armstrong J.R., *Chip-level modeling with VHDL*. Prentice Hall, Eglewood Cliffs, New Jersey 1989.

[2] Veit H., H., *A Contribution to The Generation of Test for Digital Circuits Described by Their Behavoir*. Doktor-Ingenieurs genehmigte 1992.

[3] Pla V., Santucci J.F., Giambiasi N., *On the Modeling and Testing of VHDL Behavioral Descriptions of Sequentials Circuits*. Proc. Eurodac/Euro VHDL 1993.

[4] H. Töpfer, Hümmer H. D., Automated Testcase Selection for Simulation Based VHDL Design Verification, Second Workshop on Hierarchical Test Generation, September 25-26, 1995, Duisburg, Germany.

A wind flow model over boundary layer

Thierry MARCELLI Jacques-Henri BALBI

Université de Corse, SDEM URA-CNRS n° 2053, Campus Grossetti, 20250 Corte
Tel. : (33) 04 95 45 00 73 ; FAX : (33) 04 95 61 05 51
e-mail : {marcelli,balbi}@univ-corse.fr

Abstract

In this study, the aim is to simulate a wind flow using a basic model for coupling with a fire spread model. In the first part, we consider the Euler equations derived from the Navier-Stokes equations thanks to some hypotheses. In the second part, we present the method of the characteristics with the associated equations, and a numerical outflow boundary condition. Finally, we present some results of simulation in one and two dimensional cases.

1 Introduction

This study is a part of a project which consists in creating a forest fire simulator in order to help firefighters facing a fire.

In this way, our team has elaborated two fire spread models [6]. A phenomenological one which was inspired from diffusion-reaction equations and a more fundamental one consisting in flame spread modelling. This last model requires when dealing with wind condition the knowledge of local wind. The aim of the present study is to elaborate a wind flow model as simple as possible (without considering the coupling with the fire) which is beyond the scope of this paper. Hence, we use the classical two-dimensional Navier-Stokes equations

$$(\mathcal{NS})\begin{cases}\dfrac{\partial\rho}{\partial t}+\operatorname{div}(\rho\vec{u})=0\\[2ex]\dfrac{\partial\vec{u}}{\partial t}+\left(\vec{u}\cdot\vec{\nabla}\right)\vec{u}=-\dfrac{1}{\rho}\vec{\nabla}p+\dfrac{\mu}{\rho}\Delta\vec{u}\end{cases}\qquad(1)$$

2 Modelling hypotheses

In order to neglect the viscous effects, we consider the flow above the vegetation, as to define a no-slip boundary condition. Thus, the boundary layer effects are neglected.

The boundary layer concept [2] The experimental observation shows that the wind flow can be split up in two zones :

- a boundary layer near the solid surfaces and in which the viscous forces play an important role,
- an domain out of the boundary layer in which the moving fluid can be considering like inviscid fluid.

The viscous stress are important in the boundary layer. In the external domain, the velocity cross gradient becomes weak and the viscous stress are neglected. Hence, we can compute the wind flow like an inviscid one.

On the wall, the normal velocity component is zero, the tangential one takes arbitrary values. This means that the mass rate through the wall is zero. Now, we consider the new solid surface like being the surface over the boundary layer.

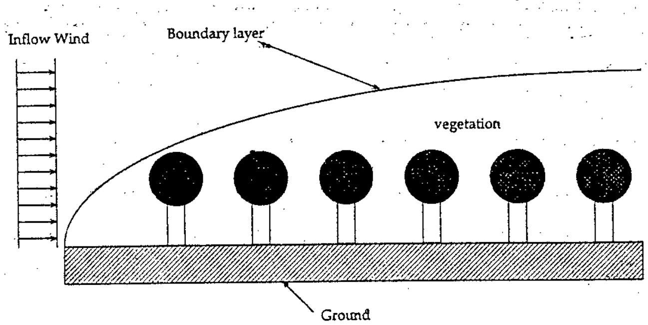


Figure 1: Boundary layer.

The set of equations is not closed, an equation is missing. This last one is given by the fluid state equation. Hence, we consider the inviscid fluid, we can assume that the fluid transformations are reversible (no friction) [4]. Like some other writers [7], we do the hypothesis of weakly compressible flow :

$$\gamma p = \rho c^2 , \tag{2}$$

where ρ is the fluid density, c the sound celerity and $\gamma = \frac{c_p}{c_v}$.

Those hypotheses lead us to the following Euler equations

$$(\mathcal{E}) \begin{cases} \dfrac{\partial \rho}{\partial t} + \operatorname{div}(\rho \vec{u}) = 0 \\ \\ \dfrac{\partial \vec{u}}{\partial t} + \left(\vec{u} \cdot \vec{\nabla}\right) \vec{u} = -\dfrac{1}{\rho} \vec{\nabla} p \end{cases} \tag{3}$$

3 Numerical method

To solve this problem (2) + (3),we use the method of the characteristics which seems to be adapted to this kind of problem [1][5]. This method consists in following the particles along their trajectory.

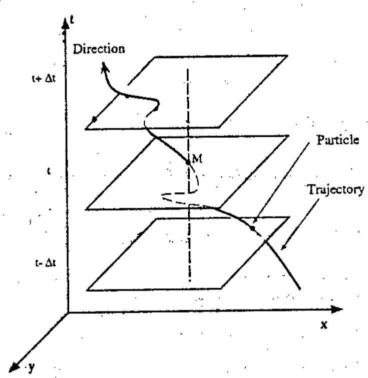


Figure 2: Trajectory of a particle.

3.1 Trajectory equation

We write $\vec{X}(t)$ the vector position. We have

$$\frac{d\vec{X}(t)}{dt} = \vec{u}\,. \tag{4}$$

Using an euler explicit scheme, we obtain

$$\vec{X}(t - \Delta t) = \vec{X}(t) - \Delta t\, \vec{u}(\vec{X}(t - \Delta t), t - \Delta t)\,. \tag{5}$$

3.2 Continuity equation

We use the total derivative of the scalar function ϕ

$$\frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} + \left(\vec{u}.\vec{\nabla}\right) \phi$$

and we can write from (3)

$$\frac{D \log \rho}{Dt} = -\text{div } (\vec{u}) \,. \tag{6}$$

Using an euler implicit scheme, (6) becomes

$$\log \rho(\vec{X}(t), t) = \log \rho(\vec{X}(t - \Delta t), t - \Delta t) - \Delta t \,\text{div } (\vec{u}(\vec{X}(t), t)) \,. \tag{7}$$

3.3 Momentum equation

As previously we use the total derivative for (3) and taking into account (2), we obtain

$$\frac{D\vec{u}}{Dt} = -\frac{c^2}{\gamma} \vec{\nabla} (\log \rho) \,. \tag{8}$$

Using an euler implicit scheme, (8) becomes

$$\vec{u}(\vec{X}(t), t) = \vec{u}(\vec{X}(t - \Delta t), t - \Delta t) - \frac{c^2}{\gamma} \Delta t \, \vec{\nabla}(\log \rho(\vec{X}(t), t)) \,. \tag{9}$$

The equations (9) and (7) show that it is necessary to know variables at $\vec{X}(t-\Delta t)$ at the time $(t-\Delta t)$. Also, we do an interpolation at $\vec{X}(t-\Delta t)$ thanks to the equation (5).

4 Simulation results

Here, we do some simulations in one and two dimensional cases. The results of those simulations are only qualitative.

4.1 Simulation characteristics

We have a fluid at rest on which we impose a inflow boundary condition : a speed flow u_e. The parameters values are given in the following table.

u_e $(m.s^{-1})$	c $(m.s^{-1})$	ρ $(kg.m^{-3})$	dx (m)	dy (m)	Δt (s)
10	331	1, 2	10	10	10^{-3}

Table 1: 1D simulations data

4.2 Numerical boundary condition

We have applied the Sommerfeld condition (an acoustical condition) which specifies that the wave is absorbed at infinite, i.e. the outflow boundary condition is a propagative condition [3].

4.3 One-dimensional case

The domain is discretized in one-hundred nodes. The delay of the simulation is 20 s. In the following figures, we represent the evolution of different points into the domain. We note that the system has converged on a steady solution.

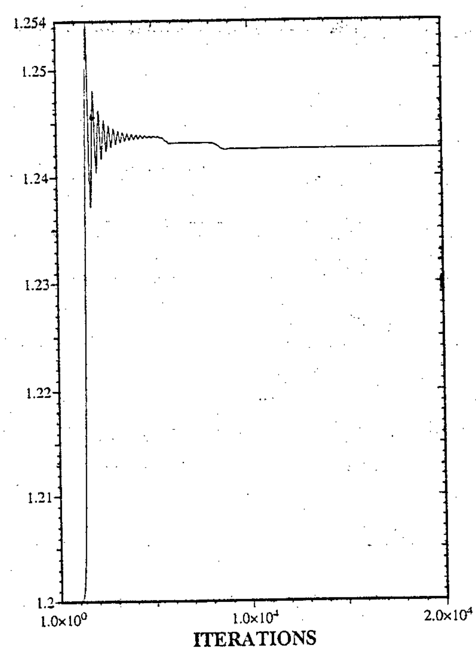


Figure 3: Density evolution at point 39

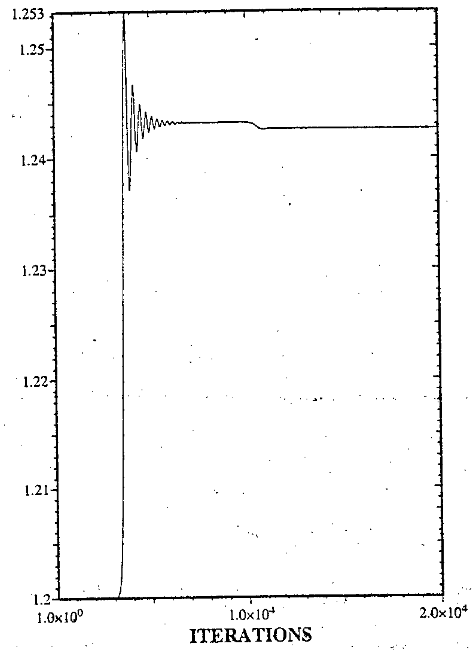


Figure 4: Density evolution at point 99

On the curves, we note a peak which represents the shock between the fluid at rest and the moving one. And some fluctuations, the value of density has converged.

4.4 Two-dimensional case

The two-dimensional study is similar for the resolution process. The parameters values are given in the following table.

u_e $(m.s^{-1})$	v_e $(m.s^{-1})$	c $(m.s^{-1})$	ρ $(kg.m^{-3})$	dx (m)	dy (m)	Δt (s)
10	0	331	1, 2	1	1	10^{-3}

Table 2: 2D simulations data

We have computed two cases :

1. wind flow without an obstacle,

2. wind flow with an obstacle.

4.4.1 Wind flow without obstacle

In two-dimensional case, (3) becomes

$$
(\mathcal{E}2)\begin{cases}
\dfrac{\partial \rho}{\partial t} + \left(\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y}\right) = 0 \\[2ex]
\dfrac{\partial u}{\partial t} + \left(u\dfrac{\partial u}{\partial x} + v\dfrac{\partial u}{\partial y}\right) = -\dfrac{c^2}{\gamma}\dfrac{\partial \theta}{\partial x} \\[2ex]
\dfrac{\partial v}{\partial t} + \left(u\dfrac{\partial v}{\partial x} + v\dfrac{\partial v}{\partial y}\right) = -\dfrac{c^2}{\gamma}\dfrac{\partial \theta}{\partial y}
\end{cases} \tag{10}
$$

where $\theta = \log \rho$.

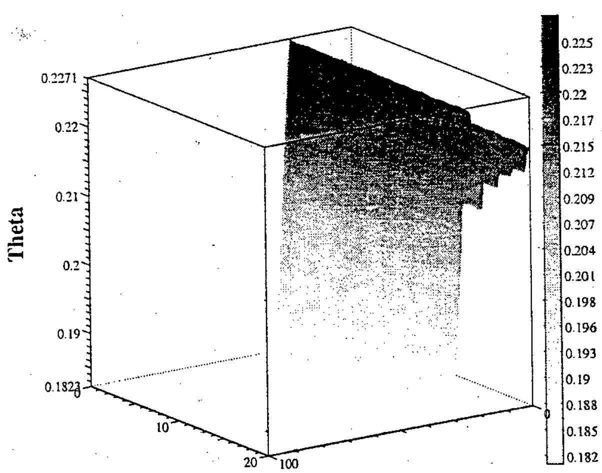


Figure 5: Pressure at $t = 0, 1s$.

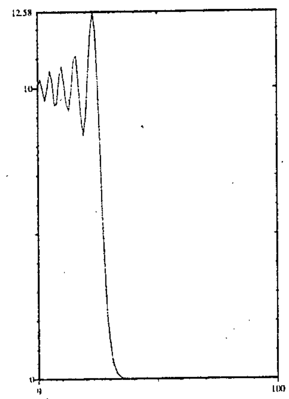


Figure 6: Velocity profile at $t = 0,1s$.

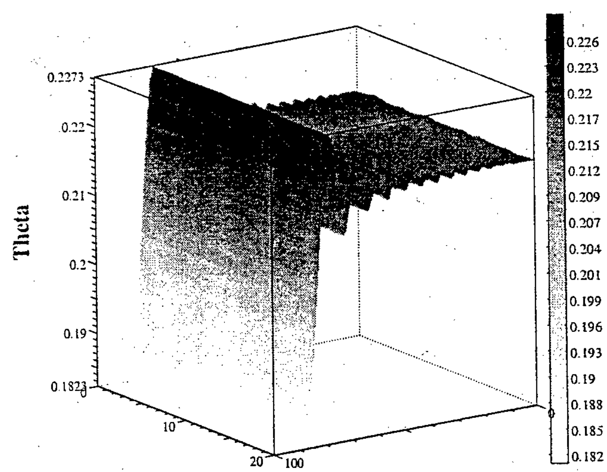


Figure 7: Pressure at $t = 3s$.

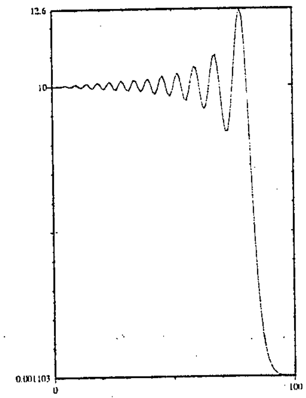


Figure 8: Velocity profile at $t = 0,3s$.

4.5 Wind flow with obstacle

We note over the obstacle an acceleration of the velocity and behind it a recirculation. At downstream, the fluid joins the ground and becomes again like at upstream.

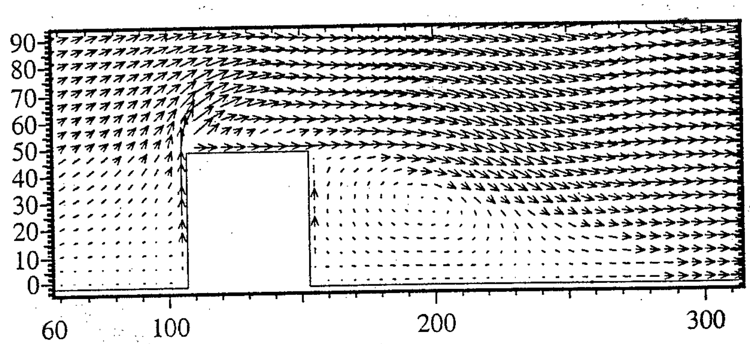


Figure 9: Velocity field around an square obstacle.

5 Conclusion

In conclusion, we have proposed a model which provides a coherent approximation for the wind velocity over the vegetation. With regard to the method of the characteristics we have noticed that this method is sensitive to the time step and the mesh size. Moreover, it should be pointed out the necessity to use a numerical boundary condition at the outflow when coupling this method with the finite difference one.

Thanks to this step we will now investigate to couple the wind flow model presented here with our fire spread model.

References

[1] M. BERCOVIER, O. PIRONNEAU, and V. SASTRI. Finite elements and characteristics for some parabolic-hyperbolic problems. *Appl. Math. Modelling*, 7:89–96, 1983.

[2] S. CANDEL. *Mécanique des fluides*. Dunod Université, 1990.

[3] J. H. CHEN. Numerical boundary conditions and computationnal modes. *Journal of Computionnal Physics*, 13:522 – 535, 1973.

[4] R. COMOLET. *Mécanique expérimentale des fluides. Tome I : Statique et Dynamique des fluides non visqueux.* Masson, 1990.

[5] J. DOUGLAS and T. F. RUSSEL. Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures. SIAM *J. Numerical Analysis*, 19(5):871–885, 1982.

[6] P.-A. SANTONI. *Propagation de feux de forêt : Modélisation dynamique et résolution numérique, validation sur des feux de litière.* PhD thesis, Université de Corse, 1996.

[7] C. C. S. SONG and M. YUAN. A weakly compressible flow model and rapid convergence methods. *Journal of Fluids Engeneering*, 110:441–445, 1988.

Discrete event modelling of an energetic system through a hierarchical approach

Christophe HALUPKA
Université de Corse, URA CNRS 2053, quartier Grossetti, BP 52, F-20250 CORTE, FRANCE
Phone: (+33) 4 95 45 00 71, Fax: (+33) 4 95 61 05 51
halupka@univ-corse.fr

I Intro

This paper deals with an original approach for defining a general framework for modelling and simulation of complex systems. This approach is based on hierarchical multi-views modelling concepts and powerful object oriented features in order to define an efficient and evaluative simulation tool.

The framework allows to express a system according to: (i) behavioural and structural views and (ii) three kinds of hierarchies (abstraction hierarchy, temporal hierarchy and description hierarchy)

II Presentation of the problem

II.A The energetic problem

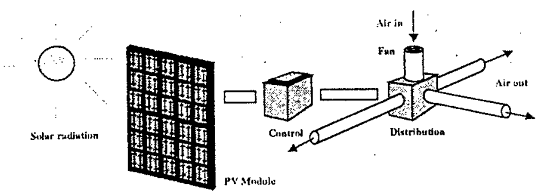


figure 1: the energetic problem

The system is a system of air renewal fed by a photovoltaic source. The modelling should allow to anticipate the exit air flow as a function of the irradiation, the temperature of the photovoltaïc item and pressures losses induced by air distribution system.

The aim of this modelling will be to choose, in an manufacturer equipment list, these that adapt the better and quoted conditions previously.

We have modelled our system into two worldviews : the structural one and the behavioural. The system is physically divided in two parts which are the PV module and the air distribution system. The behavioural part is described under.

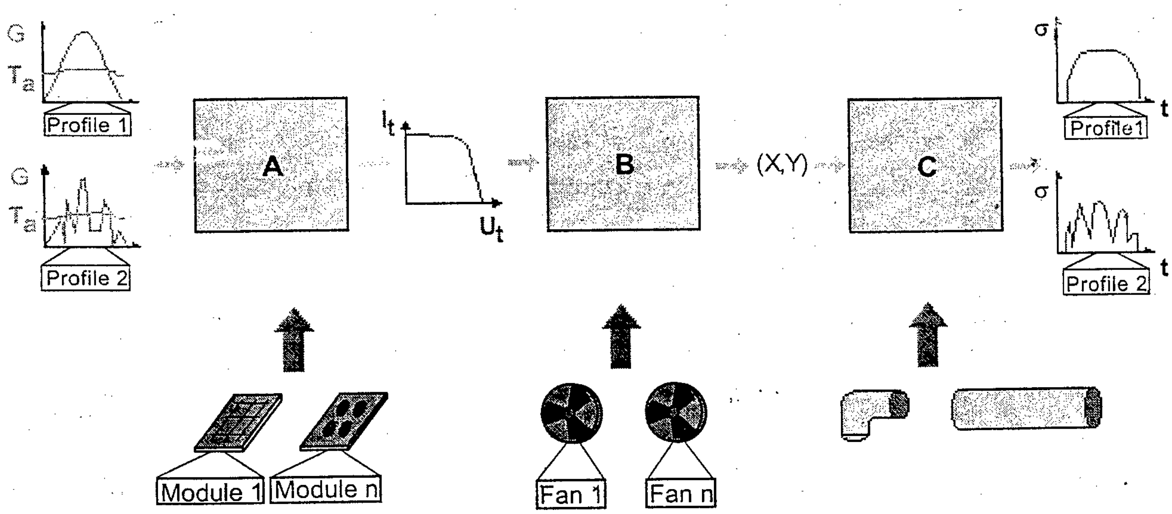


figure 2: behavioural description of the problem

Entries of the model are two curves, (irradiation curve and temperature curve), the first component produces a curve $f(I_t, U_t)$ as output. Different profiles are tested to simulate different types of day. It is possible to replace modules to observe the model evolution.

The second component takes one curve entry $f(I_t, U_t)$ and proposes as output, a couple (x, y) of a point which represents the intersection between the entry curve and fan characteristics.

Finally, the third component selects in fan Abac what better component distribution to choose. It takes as an entry, the couple (x, y) and the fan Abac, to make the better choice and give the better value as an output.

II.B Solving of a physic problem

A scientist observes physic phenomena to model them. By modelling, it means that scientist retains major parameters to establish and simplify relations between each parameters. This simplification is called a model.

Each model must be coupled with a control structure, called simulator, which interprets a possible behaviour of the model. The simulation produces hypothetical behavioural data of the system. Many types of simulation exist and we restrict ourselves to use the discrete event simulation type.

We have to notice, as it is said previously, that modelling and simulation are two different things. They must not be dependent. Modelling should not be restricted to one type of simulation. They must be clearly separated.

Simulation is usually done for one specific model. Methods and computer code serve once because they are developed for a specific model. Now, to describe models of different domains, we use a general methodology provided by a modelling environment. Thus, problems modelled in this environment use this general methodology. The model is well separated from the

simulation. Simulation part is automatically built from the modelling part. This automatic building is due to hierarchies used in the model.

A major step in analysis and conception system is due to the building of a model. A model is a simplified representation of structural and behavioural system. The first difficulty is to have reliable models. It means that they can give a consistent representation of the studied system. The second one is the validation of these models. This validation goes through the process of simulation. This simulation produces results which must be closest to the system responses.
Several works have been done about different systems modelling and simulation environment. Particularities of these works are only concerning specific domain problems. It is the reason why, used methodologies during models development depend on the type of studied system. Its validation inevitably need the building of different simulators or a large modification of existing ones.

We utilise a general environment allowing the use of the same methodology for analysing different domain systems, and use a generic simulation technique for automatic simulator building.

We have a modelling and simulation environment which makes a break between modelling and simulation. It automatically builds the simulator according to a number of rules, and according to the modelling part. It means that we have no need of building or changing the simulation part. The interest become really interesting for scientists who do only have to concentrate on model and thus do not have to build the associated simulator.

III Method

Simulation modelling is receiving ever more attention from a few years. Works done by Zeigler [Zeigler 1984; Zeigler 1990] on Discrete Event System Specification (DEVS) have boosted the study of dynamical systems. Nowadays the DEVS mathematical system theory provides a framework for representing and studying dynamical systems. The apparition of a large discrete event models coming from all the studying domains have contributed to build simulation and modelling environment.

We use a DEVS methodology environment to model our system [Aiello 1997]. Now, with the study of energetic systems, we will validate or modify this environment. To model our particular air renewal system fed by a photovoltaic source, we had to add several changes in the modelling environment.

III.A Structure of the modelling

To model a given system [Zhang 89], our approach consists on combining:

A discrete event specification system formalism [Zeigler 84, Zeigler 90] allowing to model system in a hierarchic and modular way.

A hierarchical and multi-views modelling method from C.Oussalah [Oussalah 88, Oussalah 95], allowing to add gradual complexity and to consider it in a specific way, as an expert from the specific domain.

A temporal hierarchy [Euzenat 94] allowing to keep only needed information.

The oriented object approach [Ferber 90, Meyer 88, Meyer 90] allowing to automatically generate simulation algorithms by instantiation and the use of message passing.

These four basic concepts are needed in order to develop an efficient hierarchical modelling and simulation approach :

- the abstraction hierarchy corresponding to the definition of different levels of abstraction (Oussalah *et al.* 1995) allowing to take into account the complexity of a system in a gradual way,
- the time hierarchy corresponding to the definition of different levels of temporal granularity (Euzenat 1994) allowing to take into account only the pertinent information at a given temporal level,
- the description hierarchy corresponding to the definition of different levels of description (Zeigler 1976; Zeigler 1984; Zeigler 1990) allowing to describe a system by a set of sub-systems considered at the same abstraction level,
- the multi-views approach corresponding to the definition of different views of a system (Oussalah *et al.* 1989; Oussalah *et al.* 1995) allowing to consider a system according to a structural or a behavioural view.

III.B Modelling view from the environment

In this part, we explain the model built in the environment and the relation between the system describe in II.A and the model.

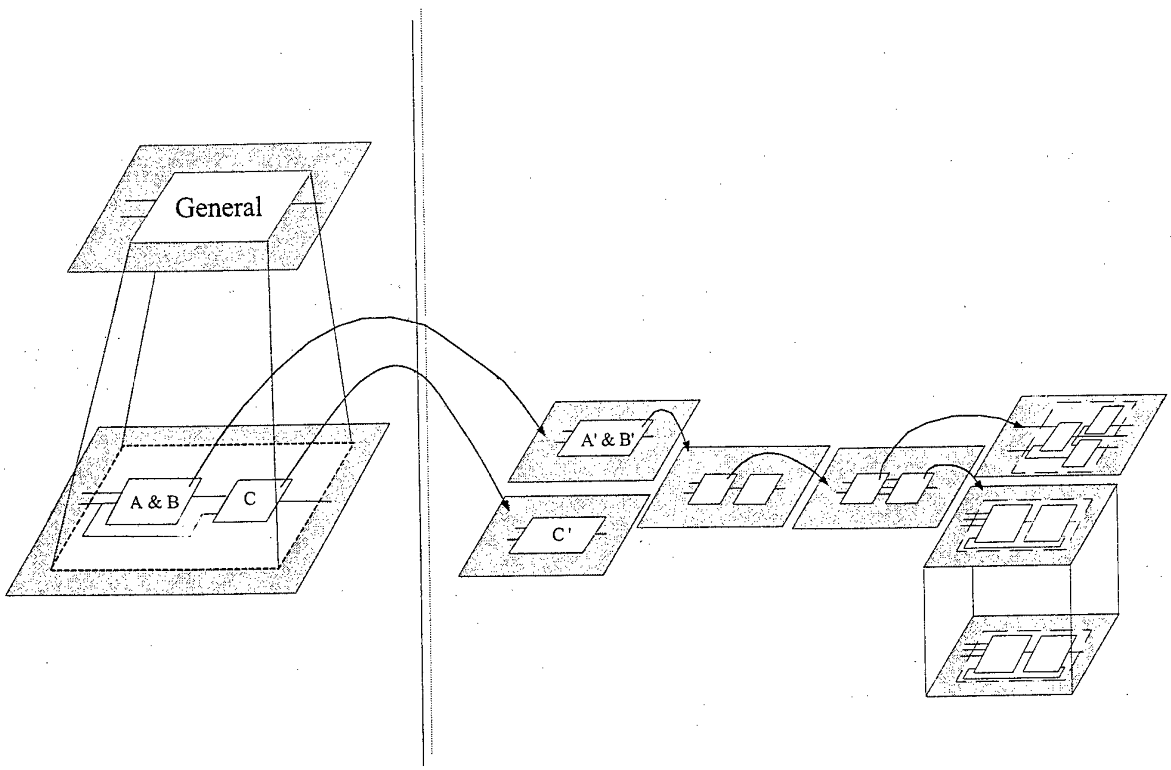


Figure 3: The energetic problem modelling according to the environment.

The figure 3 right part is the structural modelling of the problem whereas the left part is the behavioural one. In the structural part, we have a general system which have two entries (solar irradiation and temperature) and one output (pressure loss). One step under, in the abstraction hierarchy, we find a better description of the system. The subsystem A&B corresponds to A-box and B-box of the figure 2 system. They are assimilated to the solar heating system. The

subsystem C corresponds to C-box and is assimilated to the air distribution system. A&B and C subsystems do not need any abstraction decomposition.

Thus, each component is translated into the behavioural view. Simple components need only one level of description hierarchy such as C', whereas others need to be more described like A'&B'. A'&B' and C' are behavioural subsystems associated to the structural part. They do not have any correspondence with the figure 2 system. For one component we need to change the temporal hierarchy to solve correctly the problem. This temporal changing view is not correctly associated to our problem but the concept is working. This concept has many common points with the temporal concept but is a bit different. The temporal concept uses a uniform time decomposition whereas our decomposition concept is using a non uniform one. For example, we decompose one minute in sixty seconds. This is a uniform decomposition. It means, each decomposition segment has the same length whereas in our case, we decompose the set in a non uniform way. A temporal example could be the study of the only third second and nineteenth one. Here, we change the temporal view but a non uniform way. Changes done in the environment will be generalised and will be better explained in a future communication.

IV Results & conclusion

What was the limit of the modelling environment ?

For the modelling of this particular energetic problem, we have encountered three main problems: non temporal convergence, entry synchronisation and the fact of computing each couple of a two variable function.

The non temporal convergence was a problem of rising a convergence at the same time level. The problem of entries synchronisation was encountered during the execution of the transition functions. The problem generated by the two variable function was to calculate an expression , function of a variable parameter, for each time step.

Results found during this simulation were the attempted results. We have had to modify the environment to model and solve correctly our model. The general problems that we have pointed out will become more general in the environment. This will do the object of another publication.

References

Aiello A. 1997. "*Environnement orienté objet de modélisation et de simulation et de simulation à événements discrets de systèmes complexes*". Ph. D. Thesis. Université de Corse.

Euzenat J. 1994. "*Granularité dans les représentations spatio-temporelles*". Technical Report n° 2242. INRIA Rhône-Alpe, (April).

Ferber J. 1990 "*conception et programmation par objets*", Ed Hermes.

Meyer B. 1988 "*Object Oriented Software Construction*", Prentice Hall.

Meyer B. 1990 "*Conception et programmation par objets*", InterEditions, Paris.

Oussalah C. 1988. "*Modèles hiérarchisés/multi-vues pour le support de raisonnement dans les domaines techniques*", Research Report Université Aix-Marseille III.

Oussalah C. ; Giambiasi N. ; Lbath R. 1989. "*A framework for modeling and linking the structure and the behavior of a system*". In P. Born, Artificial Intelligence in Scientific Computation : Towards Second Generation Systems. J.C. Baltzer AG, Scientific Publish. Comp. Basel-Switzerland .

Oussalah C. ; Magnan M. ; Talens G. 1995. "*EMOST : un environnement de modélisation par objets de systèmes techniques*". Génie Logiciel, N°37, Sept., 31-44.

Zeigler B.P. 1976. "*Theory of Modelling and Simulation*". Wiley, New-York, .

Zeigler B.P. 1984. "*Multifacetted Modelling and Discrete Event Simulation*". Academic Press.

Zeigler B.P. 1990. "*Object-Oriented Simulation with Hierarchical, Modular Models, Intelligent Agents and Endomorphic Systems*". Academic Press.

Zhang G. and Zeigler B.P. 1989. "*The System Entity structure: Knowledge Representation for simulation modelling and design*", Artificial Intelligence, Simulation and Modelling, Ed Laurence E. Widman, Kenneth A. Loparo, Norman R. Nielsen, Wiley Interscience Publication, pp. 47-73, 1989.

NOT SO DISTANT DISTANCE LEARNING

Barbara Dziurla-Rucinska and Kent Chamberlin
http://www.ece.unh.edu/dal/lab/brvitae.htm
http://www.ece.unh.edu/faculty/chmbrln.htm
University of New Hampshire, Durham NH 03824, U.S.A.

Abstract

A pilot graduate-level course in electromagnetics has been taught at The University of New Hampshire in an effort to assess the effectiveness of using Internet technology to teach students both on and off campus. The impetus for the pilot was to provide graduate education opportunities to students for whom commuting to campus is not a viable option. The software used for this pilot is commercially available, and requires sufficiently small bandwidth so that class participants can use telephone modems. The intent of this paper is to provide the reader with an overview of the degree of effort and expense associated with initiating and executing synchronous, small-course delivery using the Internet. Further, evaluations of course effectiveness from both faculty and students is presented to show how this delivery method compares with traditional teaching techniques.

i. ACADEMIA OF DISTANCE LEARNING

Distance learning is commonly understood as teaching where students and a teacher are not required to share the same time and space in order to exercise a learning experience [1]. This relatively old concept was originated by learning through correspondence and has been revitalized because of the explosion of multimedia, television, and Internet [2], [3]. Although a precise characterization of the current status of distance learning is very difficult, a WEB search allows a reader to discover several major categories of approaches.

First, there exists a large group of "projects and initiatives" describing emerging concepts and ongoing projects in distance learning. An example can be the *Open to Europe* project at http://www.salford.ac.uk/iti/ote/, an exciting collaborative venture between 17 universities located in 10 European countries. Its central theme is the concept of "Virtual Mobility", making use of electronic networks to achieve strong collaboration between institutions in relation to project working.

Virtual universities and institutions offering distance learning services represent the second distinguishable category. Interestingly enough, the commercial aspect is not emphasized in this category, at least not directly, as opposed to services offered by virtual universities of the second category and colleges whose existence is determined solely by profit (the third category). Examples are *the IBM Global Campus*, available at http://www.hied.ibm.com/igc/facts.html and The Medical College of Virginia, the Pharmacotherapy Interactive Learning System (PILS), at http://www.software.ibm.com/is/dig-lib/vcupic.htm . The IBM Global Campus constitutes a breakthrough on education and serves as a business model that helps colleges, universities and other post secondary institutions redesign learning, teaching

and administrative functions. IBM Global Campus provides a combination of advanced technologies, network computing solutions, applications, consulting and services tailored to enable colleges and universities keep expand their offerings to new groups of students. The Pharmacotherapy Interactive Learning System is an example of a patient case-based method of teaching therapeutics. In the School of Pharmacy, faculty members are using IBM Digital Library technology to create a computer-based learning system where students can build a comprehensive patient file.

The last category is a collection of opinions, papers, and instructional materials available on the WEB, and browsing it without intelligent search engines becomes a tedious and laborious task.

The paradigm of distance learning has added a new dimension to research in education. The summary of effort in this field is concisely provided in [4]. Two extreme approaches appear to be emerging: (1) conservative, and (2) aggressive. The conservative approach considers distance learning as a natural extension of traditional teaching environment which requires the unity of teacher and learner in both space and time. This approach is known as Desmond Keegan's theory of distance learning. As a result, distance learning is implemented using traditional schema as much as possible. The aggressive method calls for innovation and separation of traditional and distance learning [5]. The latter is perceived by a substantial percentage of teaching faculty as a threat leading to potential unemployment. The new role of a teacher is not the only challenge in distance learning. A dominant one is the required technology, which has several important ingredients. Among them institutional commitments, the cost of initial investment, breaking of institutional barriers, and the growing cost of maintenance are the most important.

ii. NEW HAMPSHIRE HERITAGE IN DISTANCE LEARNING

New Hampshire Public Television (NHPTV) has been in education for almost 40 years [6]. An example of an early, painstaking effort to experiment with the Internet was the **Virtual Classroom** [7] exercise conducted in 1993 between the University of New Hampshire and the Technical University of Budapest. The objective of this endeavor was to develop a prototype software to support the concept of a virtual classroom. The virtual classroom experiment started with student familiarization of the supporting software followed by an instructional process. For prototyping purposes a lecture on *Total Quality Management (TQM)* was selected since it was assumed that volunteering Hungarian and American students, who were electrical engineers, had not been exposed to this topic before. The remaining educational process was no different from that experienced in a real classroom with the exception of a large distance and a prolonged lecture time requiring several days. The process began with a reading assignment and the instructional material. It was transmitted over the Internet followed by a question and clarification stage, and homework assignment. Next, there was a second clarification stage followed by a group organization stage (mixed teams of Hungarian and American students working together), a task assignment stage which was completed by submitting the answer, and finally an evaluation stage. The experiment was evaluated by all participants, both faculty and students, which led not only to the discovery of technical limitations of the experiment but also to the identification of sociological roadblocks..

The software, called *VOICE* (*Virtual Organizations through Internetworked Collaborative Environments*) used in the experiment was developed in-house by two UNH graduate students, Werner E. Niebel and David Brusseau. Similar attempts available at that time such as BESTnet and developed by Digital Equipment Corporation, were found unacceptable since they did not support TCP/IP protocol. The VOICE software was based on a single server with a file management subsystem embedded into the security layer. It prevented concurrent updates to files through a series of locking mechanisms. Additional functions included communication and data management as well as a security scheme for files determined by a file's creator. VOICE had a set of four menus: *Query Menu* allowed a user to recognize the usage status of the system; *Mail Menu* distributed messages to other users including broadcasting; *File Menu* handled file transfer, viewing files, editing files, and privileged changes; *Administration Menu* allowed the Project Technical Support staff to change the system configuration and the users to change their passwords etc.

As an anecdote, let us bring a personal note. One of UNH professors during a seminar in Hungary while discussing the VOICE experiment had a conversation with the audience which led to several fully valid scenarios for virtual classrooms. Some of them do not require the presence of a professor, others do not require the presence of students, and in an extreme case there may be neither students nor professors present and the lecture can continue.

iii. DISTANCE LEARNING IN NEW HAMPSHIRE: CURRENT STATUS

This graduate-level course has been taught over the Internet during Spring semester, 1998. While there will be a few meetings on the Durham campus, the lecture material was presented electronically on Mondays, Wednesdays, and Fridays from 9:10-10:00 AM. Any on-campus meetings were scheduled so as to have the least negative impact on working individuals.

Although a software license for thirty students was purchased, this first pilot course was limited to an enrollment of ten so that the demands on available resources would be kept at manageable levels. Further, the students participating were quite facile with the computer, which proved to be an asset when technical difficulties did arise.

Class preparation time was perhaps the most demanding part of this pilot. Essentially, it was a full-time job to prepare for and execute the pilot course (along with other non-teaching faculty duties), even though the instructor could have given the course in a traditional manner without additional preparation. Extensive preparation time was required in spite of the fact that an artist and typist were available to help in the effort. The instructor found that it was easier and quicker to prepare class slides himself, rather than to instruct others in generating those slides. The fortunate aspect of this is that once prepared, the course materials can be used in subsequent teaching of the course without additional effort.

Homework assignments were given via e-mail as Microsoft Word file attachments to the message. Students on campus submitted their homework by hand, and those off campus sent them in using either FAX or e-mail. On student completed all of his homework assignments using Word's equation editor, a formidable task for a graduate course in Electromagnetics!

The performance of the pilot was monitored by the Teaching Excellence program at UNH through the use of questionnaires and by comments from students. Further, a good number of interested faculty "attended" the course throughout the semester. The success of the pilot, as assessed by the students themselves, their performance on exams, participating faculty and the Teaching Excellence program, is addressed in the conclusions.

Description of the Learning Environment

All students taking this class (whether on campus or off campus) have the same basic setup: a computer connected to the network and a headset with a microphone that is connected to the computer soundboard. During the class, the instructor can display PowerPoint-type slides, which may include animation, that will appear on the monitors of the students; a "white board" is also available for displaying free-hand drawings and equations should the instructor wish to elaborate on the slides or respond to questions. Students will hear the instructor's voice through their headsets.

Interaction between instructor and student can come about through three basic mechanisms. The first mechanism is equivalent to students raising their hand in class when they have a question. If students have a question when taking a class remotely, they can click on a "hand up" icon, which will notify the instructor that a student has a question. The instructor can elect to respond to the student's "raised hand" by turning the floor over to the student. In so doing, the entire class will be able to hear the voice of both the instructor and the student controlling the floor, and that student will also have control of the slides and white board. At the discretion of the instructor, control of the floor can be returned to the instructor after the question has been answered.

A second mechanism is for students who are timid about taking the floor- they can choose to write "secret" notes to the instructor that will appear only on the instructor's monitor. Using this approach, the instructor can respond to students' questions without sharing with the class the originator of the question. Also, the number and nature of such questions can give the instructor real-time feedback on how well the students are learning the material.

The instructor can get a feel for how well the class is following the lecture by using the third mechanism: posing questions to class. These questions, which can also be used for grading purposes, are displayed as multiple choice questions to the students. As the students respond to the questions, the instructor is presented with a histogram showing which of the multiple-choice answers the students are selecting. By the number of correct answers, as well as by the rate at which the students make their selections, the instructor can get a good indication as to how well the lecture material is being received.

iv. CONCLUSIONS

The overall consensus by those involved with the pilot Internet course at UNH is that it was a success. Student performance on exams and homework indicate clearly that the concepts and details delivered electronically are being understood at least as well as they would be in a traditional classroom setting. Also, while students lament the loss of the

personal interaction and spontaneity that can occur in the classroom, most felt that that loss was offset by the convenience of not having to commute to campus.
Based upon the experience with the pilot, present plans call for at least four courses to be taught in a similar manner next Fall semester. All of those courses will be at the graduate level, and many of the students will be off campus. Assuming that all goes well with those courses, additional courses will be added each semester, with the final objective being the capability to offer complete graduate programs synchronously over the Internet.

BIBLIOGRAPHY

[1] H. Perraton, "A theory for distance education, in: D. Sewart et al. (Eds), *Distance education: International perspectives*, Routledge, New York, 1988

[2] *Distance learning and courseware links*, http://www.proinet.com/proinet/text/dlclinks.html

[3] *Distance education. Self-directed learning web sites. Annotated bibliography*, http://www.webcbt.com/debiblio.htm

[4] L. Sherry, "Issues in distance learning", *International Journal of Educational Telecommunications*, No. 4, 1996

[5] L. Sherry and R. Morse, "An assessment of training needs in the use of distance education for instruction", *International Journal of Educational Telecommunications*, No. 1, 1995

[6] F. Windsor, "Distance learning in New Hampshire", *Business & Industry Association of New Hampshire*, February 1998

[7] B. Dziurla-Rucinska and A. Rucinski, "Cyberspatial university, is it just a "virtual reality?", *IEEE Potentials*, August/September 1996

[8] K. Chamberlin, EE901 Electromagnetic field theory, *Course syllabus*, http://132.177.136.51/farview/

Not So Distant Distance Learning

Barbara Dziurla Rucinska
Kent Chamberlin
University of New Hampshire,
USA

ATW'98 USA - EUROPE

ATW'98, Ajaccio, Corsica
May 20, 1998

Presented by: A. Rucinski

Outline

- Distant Learning
- Cyberspace University
- Virtual Classroom
- Far View: Using Technology to Enhance Outreach at the University of New Hampshire
- Conclusions

Distant Learning (Perraton, 88)

- Distant Learning is commonly understood as teaching where students and a teacher are not required to share the same time and space in order to exercise a learning experience
- Cyberspace University

Virtual Classroom
(Dziurla-Rucinska, 93)

- TUB-UNH Experiment
- Prior to WEB
- Custom Made Software
- Text, Dialog, Pictures
- Menu Driven
- Experiment and Its Results

Far View: Using Technology to Enhance Outreach at the University of New Hampshire

Kent Chamberlin, Professor of Electrical and Computer Engineering, UNH (1998)

What Is Far View?

- A means for *synchronous* electronic delivery of course materials to both on and off-campus students
- Equipment required is a contemporary PC with an Internet connection
- Both on and off-campus students receive the same learning experience

Advantages

- Enables UNH to offer graduate <u>programs</u> rather than occasional courses to off-campus students
- Responds to a continuing need on the part of NH industry to more effectively provide continuing education to the workforce

Advantages (continued)

- Greatly minimizes commute time to UNH for participating students
- Responds to a new need relating to Professional Engineer certification

Cost Considerations

<u>Costs</u>

- Faculty retooling
- Course development
- Software purchase
- Increase demand on computer clusters

<u>Benefits</u>

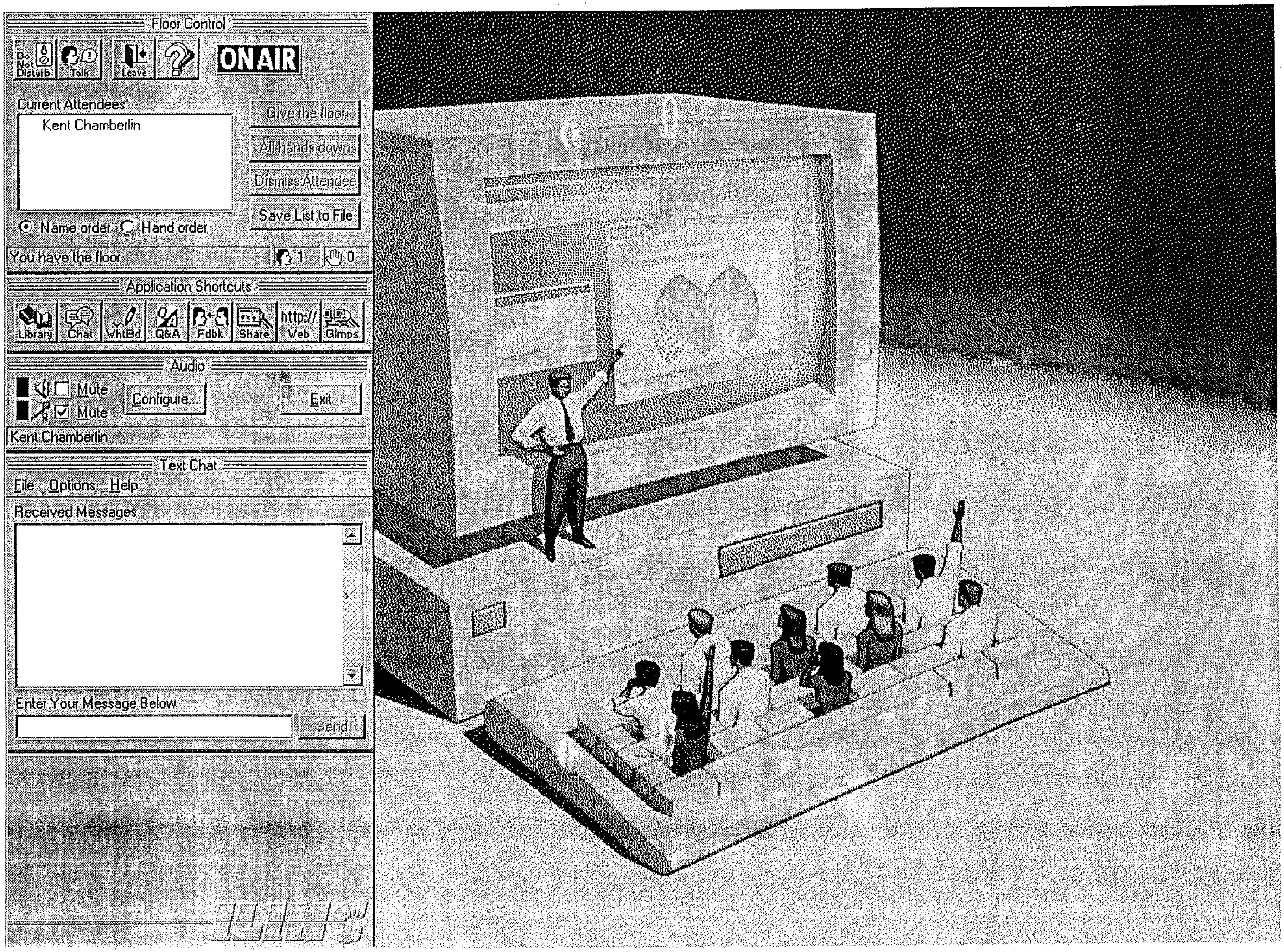
- Cost per additional student expected to be very low
- Many of the costs are one-time costs
- We may need Far View to keep our market share of students

Current Status of Far View

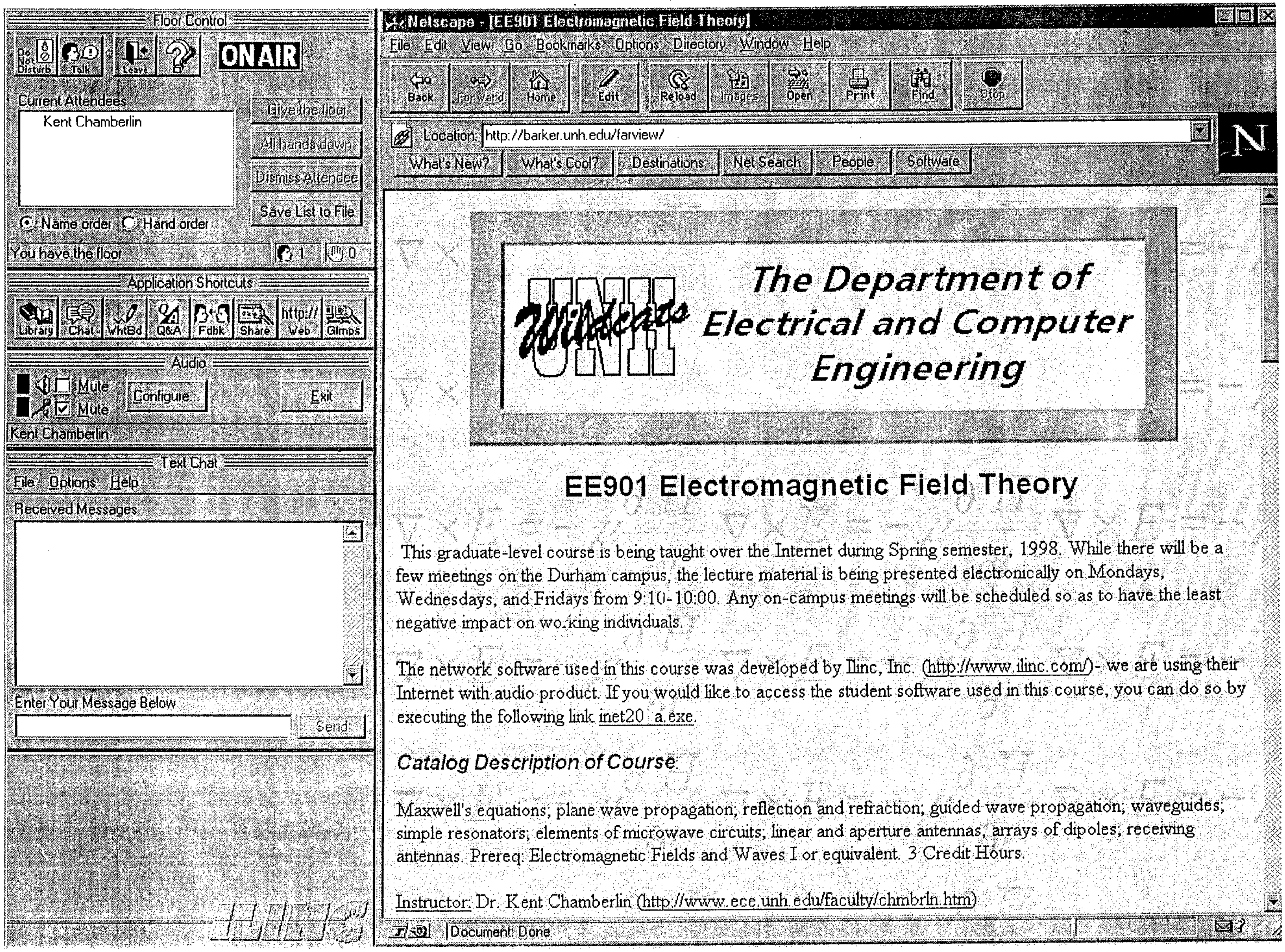
- One pilot course is being taught this semester
  - EE901 Electromagnetic Theory
- At least three graduate-level courses are planned for next Fall as part of the pilot, with additional courses scheduled for Spring 1999
- Provisions for faculty training have been made

What's It Like Taking A Far View Course?

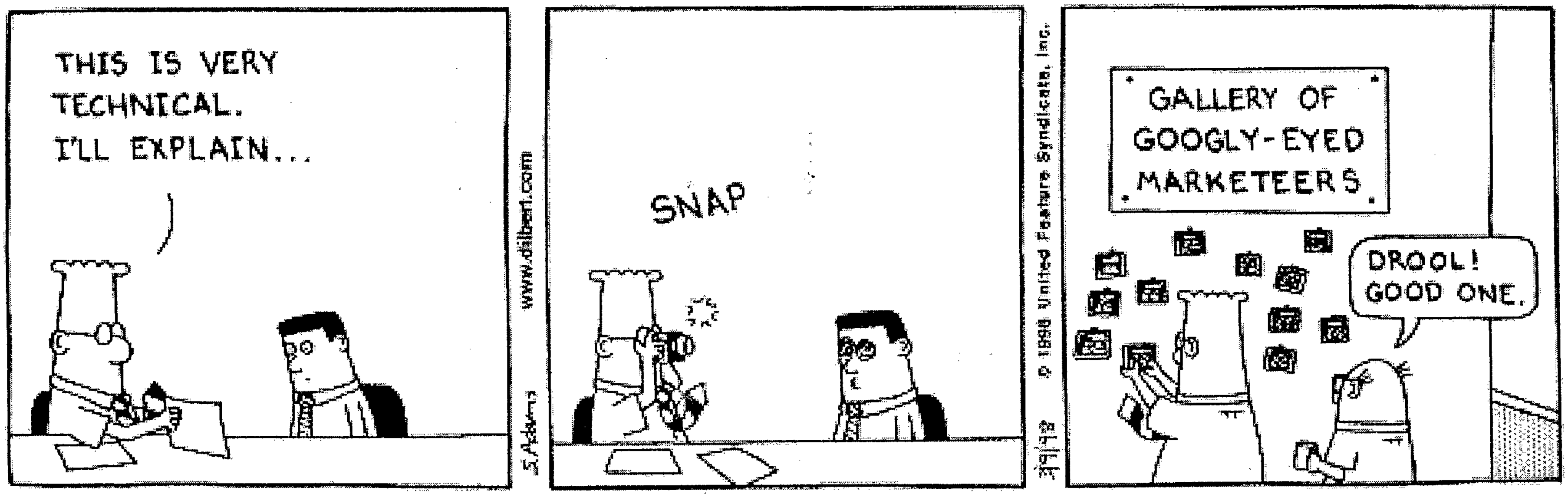
- Students and instructor sit in front of PC's equipped with headphone and a microphone
- Instructor presents material using PowerPoint type slides and white board
- Students can ask questions by "raising their hands".
- Instructor can get feedback from class via on-line quizzes.



Floor Control
Do Not Disturb
Talk
Leave
ON AIR
Current Attendees:
Kent Chamberlin
Give the floor
All hands down
Dismiss Attendee
Save List to File
Name order
Hand order
You have the floor
1
0
Application Shortcuts
Library
Chat
WhtBd
Q&A
Fdbk
Share
http://
Web
Glmps
Audio
Mute
Mute
Configure...
Exit
Kent Chamberlin
Text Chat
File
Options
Help
Received Messages
Enter Your Message Below
Send


Floor Control
ON AIR
Current Attendees
Kent Chamberlin
Give the floor
All hands down
Dismiss Attendee
Save List to File
Name order
Hand order
You have the floor
Application Shortcuts
Library
Chat
WhtBd
Q&A
Fdbk
Share
Web
Glimps
Audio
Mute
Mute
Configure...
Exit
Kent Chamberlin
Text Chat
File Options Help
Received Messages
Enter Your Message Below
Send
Netscape - [EE901 Electromagnetic Field Theory]
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Edit Reload Images Open Print Find Stop
Location: http://barker.unh.edu/farview/
What's New? What's Cool? Destinations Net Search People Software
The Department of Electrical and Computer Engineering
EE901 Electromagnetic Field Theory
This graduate-level course is being taught over the Internet during Spring semester, 1998. While there will be a few meetings on the Durham campus, the lecture material is being presented electronically on Mondays, Wednesdays, and Fridays from 9:10-10:00. Any on-campus meetings will be scheduled so as to have the least negative impact on wo.king individuals.
The network software used in this course was developed by Ilinc, Inc. (http://www.ilinc.com/)- we are using their Internet with audio product. If you would like to access the student software used in this course, you can do so by executing the following link inet20_a.exe.
Catalog Description of Course:
Maxwell's equations; plane wave propagation; reflection and refraction; guided wave propagation; waveguides; simple resonators; elements of microwave circuits; linear and aperture antennas; arrays of dipoles; receiving antennas. Prereq: Electromagnetic Fields and Waves I or equivalent. 3 Credit Hours.
Instructor: Dr. Kent Chamberlin (http://www.ece.unh.edu/faculty/chmbrln.htm)
Document: Done

Do Mondays make you Googly-Eyed?



Last Lecture

- Plane wave reflection & transmission
  - –oblique incidence
  - –Brewster angle

Today's Lecture

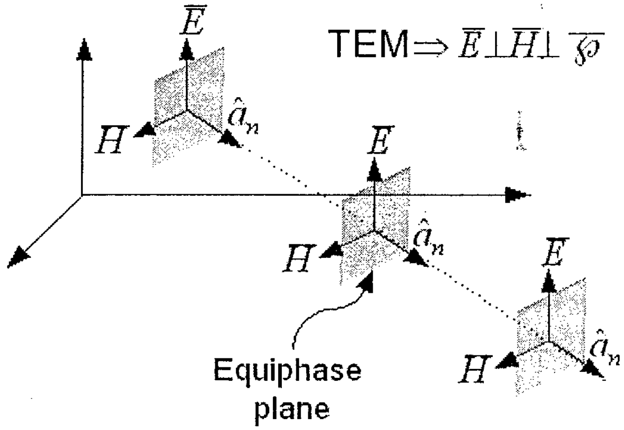
- Plane wave reflection & transmission
  - –Critical angle
    - –complex angles

Untitled - LearnLinc Whiteboard

File Edit View Grab Sharing Help

Wave Propagation and Polarization

We will be looking at wave behavior in infinite lossless media, where waves travel in transverse electromagnetic, or TEM, mode. Mode simply implies a particular field configuration.

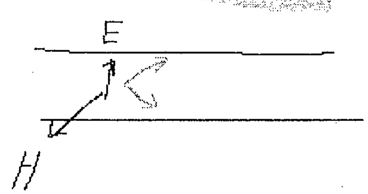


TEM$\Rightarrow \overline{E} \perp \overline{H} \perp \overline{\wp}$

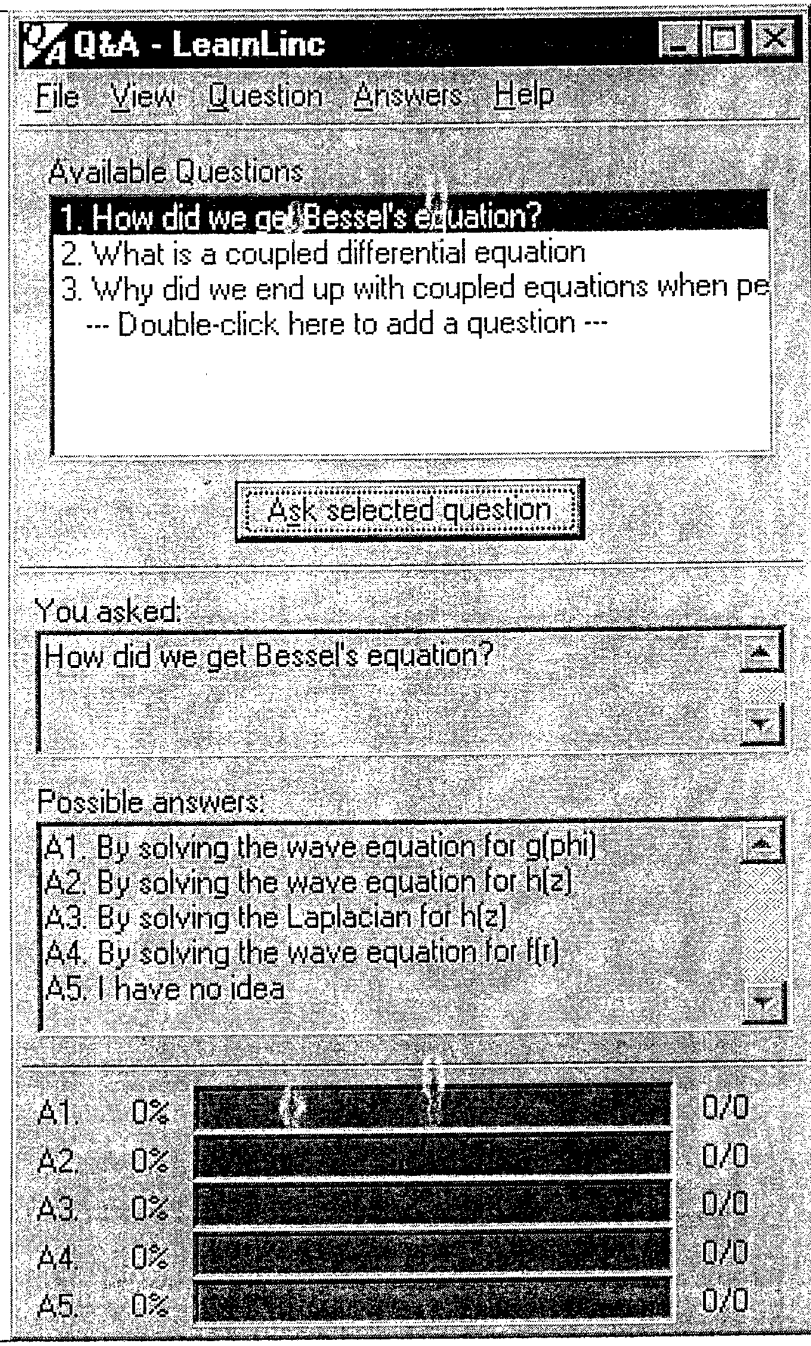
If all of the equiphase planes are parallel, the wave is a <u>plane wave</u>. If the field quantities are equal on each of the surfaces, it is a <u>uniform plane wave</u>.

The electric and magnetic fields are <u>in phase</u> in TEM mode:

$$\oint \overline{E} \times \overline{H}^* = \overline{P}_{av}$$



LearnLinc Whiteboard In sync


Q&A - LearnLinc
File View Question Answers Help
Available Questions
1. How did we get Bessel's equation?
2. What is a coupled differential equation
3. Why did we end up with coupled equations when pe
--- Double-click here to add a question ---
Ask selected question
You asked:
How did we get Bessel's equation?
Possible answers:
A1. By solving the wave equation for g(phi)
A2. By solving the wave equation for h(z)
A3. By solving the Laplacian for h(z)
A4. By solving the wave equation for f(r)
A5. I have no idea
A1. 0% 0/0
A2. 0% 0/0
A3. 0% 0/0
A4. 0% 0/0
A5. 0% 0/0

Pedagogical Pros & Cons

comments based upon on-going pilot course

Pros

- Each student gets a "front-row" experience
- Students don't take notes- can concentrate more on concepts
- Computer presentation much clearer than blackboard drawings

Cons

- Instructor does not get visual feedback
- Students miss in-class dialog
- Equipment "glitches" can disrupt classes

ATW'98
USA - EUROPE

Conclusions

- Summary
- Collaborative Engineering
- Opportunity for Corsica

Selection of an aquacole farm in Corsica using Fuzzy Logic

Pascal Muraccioli

University of Corsica
SDEM - Systèmes Dynamiques Energétiques et Mécaniques
URA CNRS 2053
BP 52 - Quartier Grossetti - 20250 Corti - France
E-mail : murracci@univ-corse.fr

INTRODUCTION

In order to determine the choice of a localization site for a marines farm, we made call to methods of multicritary aids to the decision. Indeed, for environnementales problems, making intervene many no-measurable aspects, these types of methods appear adapted especially.

This type of methods uses relations of surclassement that base their activity of estimation on the numeric functions called criterias of choice. Nevertheless, the translation of qualitative view points (or with difficulty quantifiable) in quantitative criterias is not exempt of arbitrary nature. Estimations of the various actions on such criterias are most the time blemished of imprecision.

To the numeric construction of choice criterias, one substitutes here a merely qualitative estimation procedure. The notion of qualitative criteria described of a formal way this procedure that makes call to judgments of experts. A method of aggregation of these criterias is proposed thus. However, in order to avoid all numeric transcription of estimations, the procedure of aggregation abondonne the function of utility and resorts to operators of fuzzy logic. These different stages are applied to a problem of estimations multicritary of localization of a marines farm on the Corsica coastline.

I - OBJECTIVES

In what follows, $I = \{1 \dots ,i, \dots n\}$ designates the whole of actions, and $J = \{1 \dots ,j, \dots m\}$ represents the list of points of view adopted in the problem of decision.

The idea developed in this article is to keep, at the time of the process of estimation, a common verbal ladder to all points of view. The estimation will make himself therefore more with numbers but with words. We will base our reflection on the verbal ladder used in the model SATISFACTION-REGRET (FUSTIER 1992, 1994a, 1994b, 1994c).

The presented verbal ladder is composed here of seven verbal attributes. She includes a minimal value (Inf E = forgery) and a maximal value (Sup E = true).

It exists an application:

$gj: I \rightarrow E$ (Where E is a verbal ladder permitting to value actions)

$i \rightarrow gj(i)$ = qualitative appreciation of the i action according to the j property

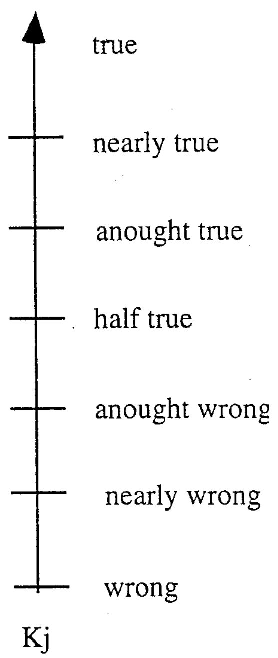
From the estimation subjective of an i object according to a characteristic j, one gets the level of truth gj(i) of the fuzzy simple proposition: the i object possesses the j property.

The profile type of an action is called qualitative pondération structure (FUSTIER 1993b). It defines itself by the application:

$$P: J \to E$$
$$j \to p(j)$$

p(j) is the qualitative weight of the j property and interprets himself as being the level of truth of the fuzzy simple proposition: the j property is important.

The ladder will use the following attributes:



The whole of actions (objects) definite I in the first chapter is constituted of the following manner: I = {i1, i2, i3, i4, i5, i6, i7}, with :

- i1 = Centuri;
- i2 = Gulf of Porto-Vecchio;
- i3 = San-Ciprianu;
- i4 = Gulf of Porto - Novo;
- i5 = Gulf of Porto;
- i6 = Gulf of Porto-Pollo;
- i7 = Shackle of Sagone.

In order to assure an optimal working of the production unit, we keep the whole of points of view following J for its installation: J = {j1, j2, j3, j4, j5, j6, j7}, with :

- j1 = Degree of protection of the site;
- j2 = Wealth faunistique / floristique of the site;
- j3 = Surface / depth of the water plan;
- j4 = Degree of accessibility to the earth;
- j5 = fundamental Statute;
- j6 = Access to an infrastructure portuaire / airport;
- j7 = tourist activity Proximity.

The different actions are valued according to their point of view. A profile type cleared himself of the various studies. One deducts some that:

- the degree of protection of the site facing inclemencies is a fundamental criteria, the estimation is true.
- the degree of wealth faunistique and floristique must be taken in amount screw to screw of impacts of an marines exploitation susceptible to modify the underlying populations; the estimation is half true.
- the availability of the water plan is valued while holding amount of three parameters: the depth, the surface and the renewal of water; the estimation is true.
- the degree of accessibility to the earth must be located close by (\0602 miles) of cages; the estimation is nearly true.
- the fundamental and authorized statute essentially concerns the accession to the maritime Public Domain; the estimation is nearly true.
- the easiness of access to an infrastructure portuaire or airport is indispensable to the viability of such a project; the estimation is true enough.
- the proximity of activities competitors can represent to enter in competition for the work of maritime public domain; the estimation is nearly true.

	j1	j2	j3	j4	j5	j6	j7
i1	NW	NW	NW	NT	NW	AW	HT
i2	T	NT	T	T	HT	T	AT
i3	AT	HT	HT	NT	HT	NT	NW
i4	NT	HT	NT	NW	HT	AW	AW
i5	HT	NW	NT	NT	NW	AW	AW
i6	NT	AT	NT	T	NT	NT	AT
i7	NT	T	T	T	NT	NT	HT
π(j)	T	HT	T	NT	NT	AT	NT

One says that the profile of an action (i) generates the satisfaction when features of the profile are high (that is to say near of Sup E) and of the regret when this same features are low (near of Inf E).

When one compares the profile of an action (i) and the profile of an ideal action (u), a difference between profiles is noted. An action can generate the satisfaction and the regret that will be meant by the verbal attributes of the E. Ladder rules governing notions of satisfaction and regret therefore are those of the fuzzy logic (FUSTIER 1992, 1994a, 1994b, 1994c).

II - QUALITATIVE CRITERIA EXTENSION

Recall: pj: qualitative criteria,
pj(i): evaluation of the i action on criteria pj.

Our objective is to determine, thanks to the method SATISFACTION-REGRET, the ordering of the seven sites preselection. We interest ourselves in this section to the qualitative criterias and rules that govern them.

1. Determination of the satisfaction

From the picture of estimationss, given in first left, we are going to define the satisfaction produced by a profile.

To arrive to these results, we apply the following rule: **sj(i) = pj(i) $\wedge$ π(j)**

	j1	j2	j3	j4	j5	j6	j7	S(i)
i1	NW	AW	NW	NT	NW	AW	HT	NT
i2	T	NT	T	T	HT	T	AT	T
i3	AT	HT	HT	NT	HT	NT	NW	NT
i4	NT	HT	NT	NW	HT	AW	AW	NT
i5	HT	NW	NT	NT	NW	AW	AW	NT
i6	NT	AT	NT	T	NT	NT	AT	NT
i7	NT	T	T	T	NT	NT	HT	T

S(i) represents the column of the global satisfaction indications, it will serve us later at the time of the global estimations: **S(i) = V [sj(i). j = 1 ...7] = V [pj(i) Ù p(j); j = 1 ...7].**

2. Count of the regret

Following stage, one calculates the regret produced by a profile:

sj(i) $<_t$ π(j) $\Rightarrow$ rj(i) is the t + 1ème element of E

t represents the echelon of the E ladder

j	j1	j2	j3	j4	j5	j6	j7	r(i)	r*(i)
rj(i1)	NT	NW	NT	W	AT	AW	AW	NT	NW
rj(i2)	W	W	W	W	AW	W	NW	AW	AT
rj(i3)	AW	W	HT	W	AW	W	AT	AT	AW
rj(i4)	NW	W	NW	AT	AW	AW	HT	AT	AW
rj(i5)	HT	AW	NW	W	AT	AW	HT	AT	AW
rj(i6)	NW	W	NW	W	W	W	NW	NW	NW
rj(i7)	NW	W	W	W	W	W	AW	AW	AT

r(i) represents the regret produced by the estimations of the object (i) according to j.

r(i) = V [rj(i) , j = 1... .7]

r * is the no-regret produced by the estimation of an i object. It is contrary r(i), he/it obeyed to the proposition the profile of the i object is not regrettable, to every i object one applies the following formula: **r(i) $<_t$ E sup $\Rightarrow$ r*(i) is the t + 1ème element of E.**

3. Global estimations of objects

The global estimation of objects is given by the level of truth of the conjunction: the estimation of the i object is source of satisfaction AND doesn't produce a regret. One applies to every i object the mathematical equation:

g(i) = min [**s(i), r*(i)**], one gets the following results:

r(i)	r*(i)	g(i)
NT	NW	NW
AW	AT	AT
AT	AW	AW
AT	AW	AW
AT	AW	AW
NW	NW	NT
AW	AT	AT

From the criteria of global estimation, one can do an ordering where objects are arranged in a decreasing order of after their global estimation. One gets the ordering of sites of most favorable to the favorable month:

I6 PS i2 I i7 PS i3 I i4 I i5 PS i1

The most favorable general estimation is gotten by the site of Porto Pollo (i6), follow-up of two Porto-Vecchios sites and Sagones (i7 and i2) confirming previous method results while being more categorical.

III - THE COMPUTER TOOLS

1. Conception of a practiced system

The practiced systems, in particular, look for to replicate the appraisal of human through the machine. In order to put in evidence the interest of the practiced system utilization (AMAT and YAHIAOUI 1996) in the setting of our problematic, we conceived three programs using data presented in the matrix of estimation of this part successively. We chose to develop these practiced systems to use the language of programming Prolog IV (PrologIA 1995, DELAHAYE 1988). This choice is not innocent, indeed, this language permits to conceive some evolutionary programs (MURACCIOLI 1998a), it is what we demonstrate all along this chapter.

To achieve this program, we based ourselves on the analysis achieved in the previous chapter concerning methods multicritary of aids to the decision. We translated the different information collected under shape of facts that consists in valuing every action according to a point of view considered thus. First of all we affect for every point of view a predicate. To title of example, to represent the point of view degree of protection of the site, we defined the predicate degree. In the same way, for:

- wealths faunistiqueses and floristiqueses of the site, predicate wealth,
- the surface and the depth of the site, predicate superf,
- the degree of accessibility on the ground of the site, predicate access,
- the fundamental statute of the site, the predicate statute.

Then for every action, we define from these predicates, different facts. These facts permit to value the performance of every site according to the point of view considered.

For example, in the previous chapter, for action centuri and according to the point of view degree of protection of the site, we had fixed an estimation of value 5 (the ladder of definite estimation of 0 to 20); this information is represented in our program of the following manner:

degree (' centuri',5).

Our three practiced systems are structured of the same way, with the same predicates,: every action is compared to others according to every retained view point. They differ then on the fashion of count permitting to do the ordering of actions.

2. Implementation

In the first program this count is simplest, we integrated then successively then in the second the third program different improvements.

In the second program, facts are identical. In order to take in amount the importance of points of view in the process of decision, we modified the rule whole. So to every point of view, is affected a weight reflecting the judgment of the décideur.

For an action data (X), one searches for all points of view their estimations. Before being added, these estimations are multiplied by their respective weights. The gotten sum is divided then by the whole of weights of points of view.

This program doesn't always permit to get an affirmed ordering. Indeed, he/it is not always possible to overrule the Porto-Pollo sites and Porto-Vecchio. In order to refine this ordering, we modified this program all over again while doing to vary weights assigned to the different points of view.

Thus, we succeeded, while achieving a simple modification (change of weights) to overrule the Porto-Pollo sites and Porto-Vecchio (Porto-pollo represents the optimal solution).

Sites	SE1	SE2	SE3
Centuri	3,5	3,2	3,06
Porto-Pollo	10,66	9,96	10,41
Sagone	10	9,20	9,79
Porto	6	5,79	5,13
Porto-Vecchio	10,66	9,96	9,89
San-Cipriano	9,66	8,93	9,62
Porto-Novo	7	6,75	5,79

In this picture, we present a summary of the descended results of the three programs achieved. In order to solve our problem, we achieved in a first time a practiced system (SE1) that didn't permit to get some satisfactory results (two sites being ex-aequo). This practiced system has been improven therefore (SE2) while integrating parameter weight comfortably. The program SE2 always not permitting to get some satisfactory results, it has been modified (SE3). indeed, in SE3, we changed values of weights; it permitted to overrule the two sites ex-aequos.

The interest of this approach type especially resides in the fact that the achieved programs are evolutionary. Thus, the décideur can integrate new information comfortably without having for it to modify the entirety of the program. It can thus, according to the gotten results, to refine of simple manner its analysis.

CONCLUSION

Results thus gotten of methods are presented below in the picture.

METHODES	SITES	CLASSEMENT
SATISFACTION REGRET	i6, i7, i2	i6 P i2 I i7
SE1		i6 I i2 P i7
SE2		i6 I i2 P i7
SE3		i6 P i2 P i7

The analysis of the previous picture confirms the selection of three sites: Porto-Vecchio, Porto-Pollo, Sagone (i2, i6, i7). The ordering gotten by the method SATISFACTION-REGRET confirms the selection of the three previous sites on the one hand, while achieving the following ordering: bay of Porto-pollo followed gulf of Porto-Vecchio to equality with the shackle of Sagone.

As the method SATISFACTION-REGRET, the conception of the practiced systems permits to achieve an ordering. The practiced system (SE3) contrarily to the practiced systems (SE1 and SE2) permits to get the more decided results. So of after SE3, sites presenting the best potentialities for the implantation of a navy farm are by order of decreasing preference: Porto-Pollo, Porto-Vecchio and Sagone.

Results gotten with the help of the different methods of help to the decision are homogeneous. It clears itself three particularly auspicious sites to the installation of a farm marinades with a light preference for Porto-Pollo.

Approaches thus proposed in this part permit to keep a certain suppleness of utilization while respecting some logical rules for the resolution of our problem. To the construction of criterias by functions, it seemed us preferable to use the expertise of experts, justifying the utilization of the method SATISFACTION-REGRET and the Practiced Systems.

BIBLIOGRAPHY

J.L. AMAT, G. YAHIAOU. 1996. Advanced techniques for the treatment of information. Cépaduès Editions.

J. -P. DELAHAY. 1988. Course of Prolog. Eyrolles.

B FUSTIER. 1992. Approach economic and qualitative of the decision. The model SATISFACTION-REGRET. Note of research of the CEMA n°4 University of Corsica.

B. FUSTIER. 1994a. To value the farming space with the soft methods: application to the estimation of sites. Notes of research of the CEMA n°14, University of Corsica.

B. FUSTIER. 1994b. Takes decision on the qualitative criterias. Note of research of the CEMA n°18, University of Corsica.

B. FUSTIER. 1994c. Qualitative criteria aggregation. Note of research of the CEMA n°19, University of Corsica.

Prololog IA. 1995. Manual of Prolog IV. Luminy.

B. ROY, P. BERTIER. 1971. The method ELECTRE 2. Notes of work n°142 of the scientific direction of SOWED it, Paris.

B. ROY 1985: Methodology multicritère of help to the decision. Economica, collection management, Paris.

P. VINCKE. 1989: Help multicritère to the decision. Editions Ellipses, Brussels.

A NEW TOOL FOR INTEGRATION OF RENEWABLE ENERGY SYSTEMS : G.I.S. - GEOGRAPHICAL INFORMATION SYSTEM -

M.Muselli, G.Notton, P.Poggi, A.Louche
Centre de Recherches Energies et Systèmes
Route des Sanguinaires - F-20000 Ajaccio
Tel : (33).4.95.52.41.41 - Fax : (33).4.95.52.41.42
E-mail : muselli@vignola.univ-corse.fr

ABSTRACT : The main objective of this project is to realize an integration plan for various Renewable Energy systems in remote areas using a Geographical Information System (GIS). The studied area is Corsica, an island located in the south-east of France. For a considered remote site, four systems supplying electricity are compared : a stand-alone PV/Batteries system, a hybrid PV/Batteries/Back-up generator system, an engine generator and an extension of existing electrical network. The most economic configuration has been chosen as the best solution to electrify the remote site. Physical and technical-economical processes are integrated in the GIS. This GIS is used to determine the profitability boundaries for PV systems compared to a grid extension and according to four load profiles, has led to the elaboration of an integration plan of Renewable Energies in South Corsica. The study has shown that for 60 to 90% of remote sites, a PV decentralized electricity system is the most economical way of electrification.

INTRODUCTION

A Geographical Information System (GIS) is a computerized data base which allows to integrate and to process informations coming from different sources, to elaborate afterwards strategies for development of developing countries. GIS have been developed in the late 1960s but had been installed in very few places because, on the one hand, hardware was very expensive and on the other hand, the number of softwares available on the market was limited [1]. However, with the decrease of the hardware price and the increase of computers performances, the GIS represents a good tool for political or economical authorities. As a tool box, a GIS allows planners to perform spatial analysis by using its geoprocessing or cartographic modeling functions, such as map overlay, selection and selection SQL, thematic analysis [2] etc. Among all the geoprocessing functions, the map overlay is probably the most useful tool (Figure 1).

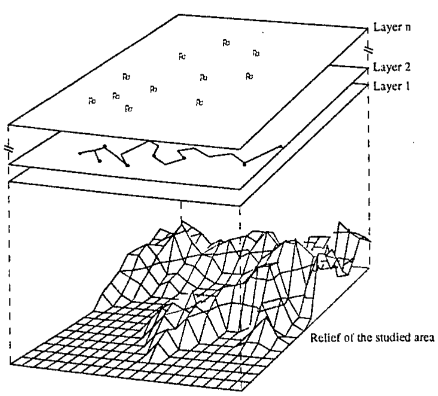


Figure 1 : Map overlay function.

The purpose of this paper is to apply the GIS methodology to renewable energy systems providing electrical energy in stand alone conditions. In a first step, the regional base is built and contains information characterizing the studied area as the solar potential, the electrical networks (medium and high voltage grids), remote sites locations ... From this regional base, computers tools are developed to size the energetic systems in relation to the electrical needs of the site. The various decentralized electricity systems are compared on the basis of a minimization of the kWh cost. At last, the main results of the study are described and the integration plan of renewable energy systems is presented.

1. PRESENTATION OF THE METHODOLOGY

Description of the studied area

The studied area is delimited by the South Corsica department. The surface represents a square of 8100 km² and the terrestrial surface about 3800 km². With about 250000 inhabitants, the island of Corsica has a mean population density equal to 27.7 inhabitants/km². But in the rural area, this density falls down to 7-10 inhabitants/km².

Description of the methodology

The flowchart of Figure 2 illustrates the complete methodology of our study.

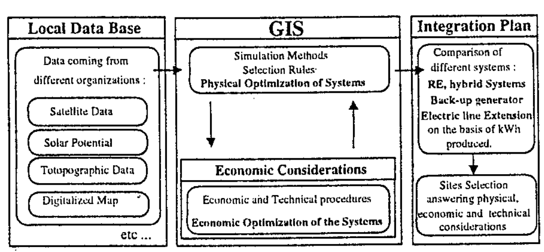


Figure 2 : Methodology used in this project.

To elaborate this GIS, a first step consists on a digitalization of data maps (RASTER mode) to create the regional data base which must be as developed and accurate as possible. Each information layer represents a particular characteristic data as solar potential, electrical network and relief provided by various sources: topographic and radar records, satellite and meteorological data, land availability ... These thematic data have the particularity to be spatially referenced. They are processed

by several computer tools to design electricity production systems : determination of components sizes for conventional and photovoltaic systems and estimation of the distance between the remote site and the nearest transformer. These various systems are compared in term of kWh production cost with an economical tool integrated in the GIS.

Construction of the regional data base

The thematic data, referenced spatially, are issued from different sources :

■ High and Medium Voltage networks : 7 mappings representing high and medium voltage grids, respectively at the scale 1/200000 and 1/50000 provided by EDF CORSE (Electricité De France).

■ Background map : 8 mappings IGN TOP25 (scale 1/25000) representing the land suitability.

■ Remote site location : By correlating the two first layers, and using the GIS, we can know the location of remote sites. This algorithm has allowed to determine 1046 individual remote sites representing 496 grouped sites (sites around a reference house in a circle of 500 meters). The distribution of the distance from the grid of these sites is presented on Figure 3. We note that 80% of sites are located at a distance inferior to 2 km from the grid what indicates that it is a matter of an electrification of proximity.

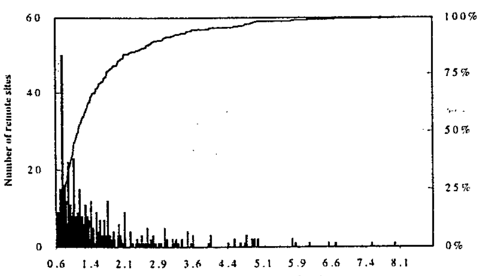


Figure 3 : Distribution of the distance between remote sites and transformer MV/LV .

■ Solar potential : In a previous study [3], we have shown the inadequacy of the pyranometric corsican network to deduce, by interpolation methods, total solar irradiations all over the island. Indeed, there is no relation from a solar radiation point of view, between two sites which are about 10 km apart. A model allowing to compute daily global irradiations on horizontal plane, based on the utilization of METEOSAT satellite derived-data had been used to cartography the solar potential [3]. The spatial resolution of satellite WEFAX images defines a pixel with a surface of about 25 km². The validation of this model has shown that the relative root mean square error between experimental and simulated daily horizontal global irradiation is inferior to 13%. The annual mean value of daily global irradiation varies in a range of 4.04 kWh.m^{-2} to 4.83 kWh.m^{-2} corresponding to a variation around 20% all over the island.

2. PHYSICAL SIZING

The various solutions envisaged to electrify remote sites are :

- photovoltaic systems characterized by its PV modules area, battery storage capacity;
- hybrid systems consisting in a PV system with an auxiliary engine generator source. Its sizing parameters are : PV modules area, battery storage capacity and nominal power of the engine generator;
- engine generator directly connected to the load via an AC/DC converter characterized by its nominal power;
- an extension of the electrical grid. The length of the medium and low voltage lines must be known.

The sizing of these systems required meteorological data as an input and electrical consumption data as an output.

Solar irradiation and load profiles

To study the behavior of photovoltaic and hybrid systems, synthetic hourly global irradiation on a tilted plane are constructed from daily global irradiation on a horizontal plane supplied by METEOSAT satellite images. This algorithm has been presented in recent works [4]. A data generator, based on an ARMA process and on the TAG model has been built to synthesize hourly global horizontal irradiation from daily ones. Then, Hay's model allows to compute hourly tilted irradiation from hourly horizontal ones with a good accuracy.

Four DC load profiles have been chosen to simulate the hourly consumption of the users : two typical annual profiles ("Day" and "Night" profiles with a consumption of 1 kWh/day); a seasonal "Low Consumption" profile based on adapted loads (1.9 kWh/day) [4] and a seasonal "Standard" profile based on the French utility EDF data (3.7 kWh/day).

Stand-alone PV systems

Concerning the behavior of the PV system, we used a method based on a system energy balance and on the storage continuity equations. To determine the PV system sizing curve (i.e. the set of characteristic couples composed of PV modules area Sj and battery storage capacity Cj), we simulated the system behavior, and according to the autonomy constraint (solar contribution SC_j equal to 100%), the battery capacity for a given module surface is obtained when, at the end of simulation period, the PV system has always supplied the load. The above-described calculations determined several autonomous PV system configurations. Thus, we used this methodology to size PV systems on each pixel on the satellite images supplied by METEOSAT.

Hybrid photovoltaic systems

The hybrid system consists in a photovoltaic subsystem with an auxiliary engine generator used as a battery charger via an AC/DC converter. From the system behavior simulation, in changing the autonomy constraint into a constraint relative to the part of the load supplied by the PV system (SC_j), we determined the set of PV system

characteristics (S_{SCj}, C_{SCj}) which leads to the solar part SC_j checking the energy balance. An engine generator characterized by its nominal power P_{SCj} is allocated to each storage size-PV modules area couple (S_{SCj}, C_{SCj}) in such a way that the hybrid energy system is globally autonomous with a percentage of SC_j % of the load supplied by the photovoltaic subsystem.. Thus, the set of couples became a set of triplets (S_{SCj}, C_{SCj}, P_{SCj}).

Engine generator system

The gasoline engine generator is directly connected to the DC load via an AC/DC converter. In these conditions, the nominal power of the gasoline engine generator is computed from the peak load power of each profile divided by the efficiency of the AC/DC converter taken equal to 90%. The calculation of the annual working time and fuel consumption are integrated in the GIS [5].

Grid extension

The extension of the grid is constituted in the model by : a medium voltage (MV) line, a MV/LV transformer and his support, one or more low voltage (LV) lines supplying the remote sites. Figure 4 describes the methodology integrated in GIS to process the extension of the existing electric network for a grouping house. The various lines lengths for both MV (D_{MV}) and LV voltages lines (d_i) are calculated.

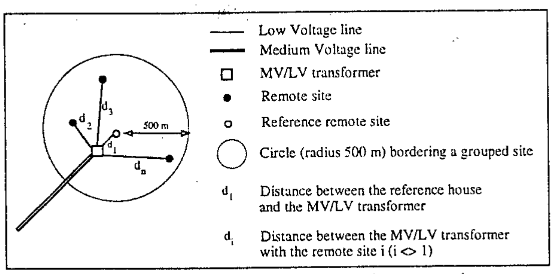


Figure 4 : Description of the grid extension procedure.

3. ECONOMICAL OPTIMIZATION

As seen previously, the physical modelisation has conduced to an great number of possible configurations for PV and Hybrid systems. An economical optimization [5] has been applied to determine between all these configurations, the system conducing to the lowest kWh cost. On each remote site, the optimized PV system and hybrid system are thus determined. The same procedure is applied to the engine generator system and to the grid extension.

Thus, for each site and each system, the cost of the electricity produced is known. Then, we can choice between these four solutions to electrify the remote site, the most profitable system. Such a comparison is presented in Figure 5 for a PV system and a grid extension.

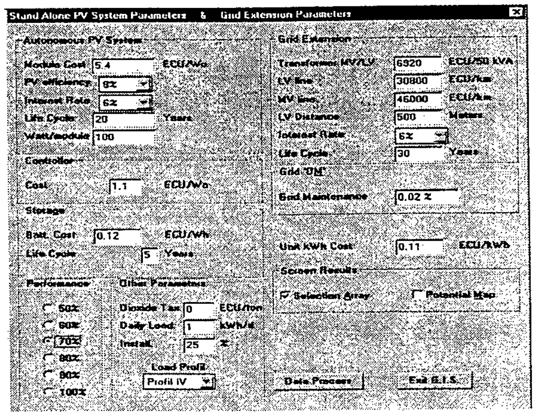


Figure 5: Interactive software for stand-alone PV systems

4. MAIN RESULTS

Profitability boundary of PV systems in remote areas

We have restricted the application of GIS, to study the profitability boundaries for PV systems in remote areas compared to the extension of the electric network. To determine these boundaries, using the GIS software, we have studied the potential of remote sites able to be electrified more economically by PV systems or hybrid systems than by a grid extension for daily energetic load increasing from 1 to 10 kWh per day.

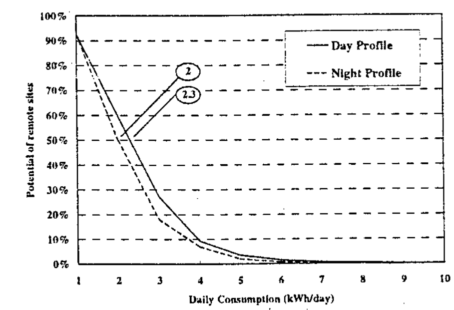


Figure 6 : Profitability boundaries for PV systems.

Results for both "Day" and "Night" profiles are plotted on Figure 6, where we see that if the needs of the remote houses are respectively 2 kWh per day with a day profile and 2.3 kWh per day with a night profile, then the photovoltaic electrification solution is more profitable than grid extension for 50% of the remote sites. We note an influence of the repartition of the load during the day on the remote sites potential. As an example, for a potential of 50% of the total remote sites, we obtain for the four profiles the profitability boundaries described in Table 1.

Boundaries (kWh/day)	Day	Low	Standard	Night
PV/Battery	2.3	2.7	0	2.0
Hybrid PV	2.9	3.6	4.0	2.6

Table 1. Insertion Boundaries versus daily load profiles

Thus, for a mean daily consumption of about 2 kWh/day and 3.5 kWh/day, we obtain a potential of 50% of remote sites susceptible to be supplied respectively by PV/Battery and PV hybrid systems. For PV/Battery configurations,

these results are in agreement with previous works concerning the study of the insertion boundaries of PV systems in remote areas [6-8].

Determination of PV potential

We look for the module area to install on each remote site to supply a given daily load for photovoltaic and hybrid systems. We have calculated a normalized module surface S_{ref} defined as the module area to install to supply a daily load equal to 1 kWh according to each load profile and for each remote site.

S_{ref} are in a range from 4 to 6 m^2/kWh/day and from 1.5 to 3 m^2/kWh/day for respectively PV/Battery and PV hybrid systems whatever the load profile is. This range takes into account the important temporal distribution of the load during the day mainly in the "night" profile. We conclude that the addition of a back-up generator in a traditional PV system decreases the surface of PV modules by a factor of 2 and the size of battery storage by a factor 4. The decrease of system size conduces to a reduction of kWh cost what explains why hybrid systems are more competitive than photovoltaic systems and that the profitability boundaries in comparison with grid extension is higher for hybrid than PV systems.

Integration of decentralized electricity production systems.

The processes previously described are integrated in GIS to study the potential of Renewable Energies in South Corsica. The four systems are compared on each site to determine the optimal electrical system producing a kWh at the minimum cost. Considering the same economical hypothesis, we have opted for 4 physical strategies based on load profiles (strategy 1 : 1 kWh according to "day" profile, strategy 2 : 1.9 kWh according to "low consumption" profile, strategy 3 : 3.7 kWh for "Standard" profile and strategy 4 : 1 kWh for "night" profile) in order to evaluate the renewable energies potential.

The potential mappings are presented on communal background map (Figure 7) for low profile and the competitiveness between various decentralized electricity production systems is summarized on Figure 8. Thus, 60 to 90% of remote sites can be supplied by hybrid PV systems. The grid extension represents then a potential varying between 10 and 40%. We can note that the increase of the daily load (by using the "Standard" profile) profits only to the grid extension. Moreover, whatever the load profile is, PV/Battery systems and back-up generators directly on the load are never competitive beside PV hybrid systems. These results are foreseeable taking into account the installation cost of electric lines in this country linked to the topographic constraint.

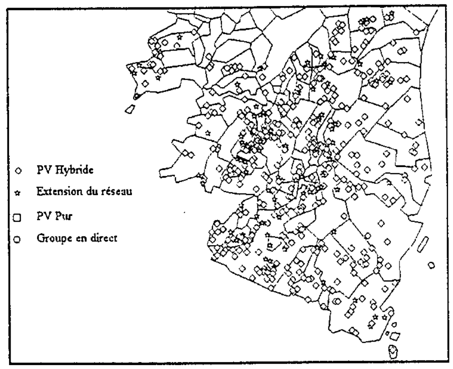


Figure 7 : Potential mapping.

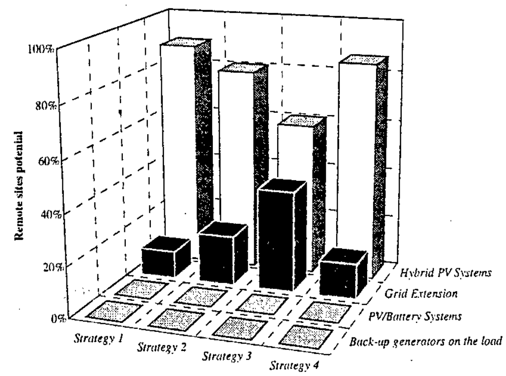


Figure 8 : Renewable energy potential

CONCLUSION

A methodology allowing to estimate the potential of Renewable Energy systems in remote areas is presented. This method is based on the utilization of a GIS and compares all the systems on the basis of a kWh produced minimization. This interactive methodology presents several advantages :

- it can be easily adapted to any site all around the world and particularly in developing countries where local data base are available
- it represents a real tool for urban and rural Renewable Energies planning.

This work shows a very high potential of remote sites (between 60 and 90% according to the strategy integrated in GIS) susceptible to be equipped by PV hybrid systems where as the grid extension represents 10 to 40% of the potential.

REFERENCES

[1] Hey, A.G.O, *Int. J. of GIS*, **5**, 5-27, (1991).
[2] Tomlin, C.D., Englewood Cliffs, (1990).
[3] Muselli, M., Notton, G., Canaletti, J.L., Louche, A., *Energy Convers. Mgmt*, **39**, ½, 1-19, (1998).
[4] Muselli, M., Thesis, University of Corsica, (1997).
[5] Notton, G., Muselli, M., Poggi, P., Louche, A., 14th EPSEC, Barcelona, (1997).
[6] Dessus, B., Devin, B., Pharabod, F., *la Houille Blanche*, **1**, (1992).
[7] Equer, B., Percebois, J., Ellipse Ed., (1993).
[8] Copetti, J.B., Macagnan, M.H., Zanesco, I., 13th EPSEC, Nice, (1995).

Toward a Recommended Practice for Architectural Description*

Walter J. Ellis
Software Process and Metrics
ellis@cse.ogi.edu

Richard F. Hilliard II
The MITRE Corporation
rh@mitre.org

Peter T. Poon
Jet Propulsion Laboratory
peter.t.poon@jpl.nasa.gov

David Rayford
Comerica Incorporated

Thomas F. Saunders
The MITRE Corporation
tfs@mitre.org

Basil Sherlund
Comerica Incorporated
she@acm.org

Ronald L. Wade
E-Systems
rwade@ix.netcom.com

Abstract

The Architecture Planning Group (APG) was chartered by IEEE's Software Engineering Standards Committee to set a direction for incorporating architectural thinking into IEEE standards. In this paper, we present a framework for architectural thinking and use it to review existing architectural practices; summarize the work of the APG and its recommendations; and provide the rationale for those recommendations.

1. Introduction

It is recognized that architecture should have a strong influence over the life cycle of a system. In the past, hardware architectural aspects were dominant, whereas software architectural integrity, when it existed, was first to be sacrificed in the course of system development.

The cost of software development and the increasing complexity of software systems has changed the relative balance. Today, software technology has matured. The practice of systems development can benefit greatly from adherence to architectural precepts, at both the systems and software systems levels. However, the concepts of architecture are not yet consistently defined and applied over the life cycle.

To address this, IEEE's Software Engineering Standards Committee (SESC) chartered an Architecture Planning Group (APG) to set the direction for incorporating architectural thinking into future IEEE standards.

The APG began its work August 19, 1995, in Montreal. At that meeting, we adopted the charter included as appendix A. Over the course of the next eight months, we met regularly to develop an Action Plan for SESC. In April 1996, that Action Plan was delivered to SESC and accepted. We have subsequently initiated an Architecture Working Group (AWG) to implement the APG recommendations.

In its deliberations, the APG set these goals for itself and any subsequent Architecture Working Group:

1. To define useful terms, principles and guidelines for the consistent application of architectural precepts to systems throughout their (full) life cycle.
2. To elaborate architectural precepts and their anticipated benefits for software products, systems and aggregated systems ("systems-of-systems" [13, 17]).
3. To provide a framework for the collection and consideration of architectural attributes and related information for use in IEEE Standards.
4. To provide a useful road map for the incorporation of architectural precepts in the generation, revision and application of IEEE standards.

It was the consensus of the APG that there is a mode of architectural thinking applicable to systems which is much wider than only software. *Every system or subsystem has an architecture.* As such, the APG work attempts to address all relevant software standards and practices and to consider interdisciplinary perspectives from a systems' life-cycle perspective. Although the IEEE SESC is software oriented, the intent is to produce standards and guides which can broadly

*In *Proceedings 2nd IEEE International Conference on Engineering of Complex Computer Systems, Montreal, Quebec, Canada, October 21–25, 1996.*

support all disciplines involved in developing systems. This "wide scope" was encouraged and affirmed by SESC and will be reflected in the framework and recommendations described below.

In order to assess the impact of the wide scope interpretation of architecture, and to identify the standards potentially impacted, the APG undertook an analysis of recent and ongoing activities in the community related to "architecture." Every effort was made to contact professional associations to determine the extent of efforts in establishing architectural standards and guidelines within the industry. Keyword searches were performed on IEEE standards as well as those of other standards setting organizations. Keyword searches were also performed using several search engines on Internet. The effort identified several hundred potential efforts. These were narrowed through a process of determining whether architecture was a primary or secondary focus. Elimination criteria consisted of documents that had architecture as a by-product or where the word was used in a descriptive sense rather than a development perspective. A list of approximately twenty standards and in-process works represents a starting point for the harmonization and coordination efforts of the Architecture Working Group with other efforts.

2. Architectural Framework

To meet our first goal of defining "useful terms, principles and guidelines," we began our work by initiating the development of a conceptual framework (or, frame of reference) for talking about architecture. In formulating the framework, we found the building metaphor – the analogy between software systems architecture and traditional building architecture quite useful [12, 21].

For perspective, we began with the common, "pre-systems" definition of the word:

> **Architecture** 1. The art or science of building or constructing edifices of any kind for human use. 2. The action or process of building. 3. Architectural work: structure, building. 4. The special method or style in accordance with which the details of the structure and ornamentation of a building are arranged. 5. Construction or structure generally.[1]

Each of these senses of the word are applicable in the systems context.

There is at present much interest and activity in system architectural thinking (for example, [9, 15, 22]). These activities would benefit from a common framework of terms and concepts. The key term is "architecture" itself. We considered the current IEEE 610.12–1990 definition of "architecture" [11]:

> **Architecture** The organizational structure of a system or component.

However, this definition was not adequate for our purposes. It has several limitations. (1) It does not distinguish an architectural level of structure for a system from the design or physical structure of that system. (2) It does not separate architectural concerns, such as the interaction of the system with other systems, from other concerns, such as details of construction. (3) It offers no basis for treating architectures as engineering objects.

Looking at a number of other definitions in the literature (for example, [4, 7, 25]), we found most suffered from these or other limitations. We settled on the following definition:

> An *architecture* is the highest-level concept of a system in its environment.

This definition addresses (1) by stating that architecture pertains to the *highest-level* of a system, distinguishing it from design and implementation concerns. As Mary Shaw admonished the First International Workshop on Architectures of Software Systems [8], "Let's not dilute the term 'architecture' by applying it to everything in sight." Our definition acknowledges (2) by reminding us that systems are situated *in their environments*, and that the architecture is in part a recognition and response to that environment. But it still does not directly address (3) — a "concept" is not particularly useful as an engineering construct. Following recent IEEE work on software design [10], we distinguish the architecture itself from *descriptions* of that architecture:

> An *architectural description* is a model — document, product or other artifact — to communicate and record a system's architecture.

This distinction allows us to retain the idea that every system has an architecture, while advocating that the *explicit* description of a system's architecture allows that system to be brought under "engineering control." The form and content of architectural descriptions may then be subject to standardization — not the architectures themselves.

There is an established engineering approach to bringing such complex entities under control, which is to "separate concerns" by identifying one or more viewpoints [23]. Therefore, our framework advocates that an architectural description be organized into well-defined pieces:

> An architectural description conveys a set of *views* each of which depicts the system by describing domain concerns.

Meszaros defines an architectural view as: "a way of looking at an architecture. Each view may have a different concept of components and relationships" [19]. In our

[1] *Oxford English Dictionary*, 1933 edition.

framework, each "way of looking" will be determined by the interests of the users of that view. A system exists in many domains — it is therefore useful to regard architecture as a multi-disciplinary practice [18]. Views offer a way to get a handle on this. Some common architectural views are listed below, reflecting typical systems concerns reflected at an architectural level of description.

- behavioral, dynamic, operational views [16, 14, 4]
- data, data flow, information views [5, 7]
- development, maintenance views [3, 6]
- distributed, network views [25, 6]
- functional, activity views [24, 25]
- logical views [14, 20]
- static views [14, 7]
- physical views [14, 4]

The selection of relevant views will usually be determined by the architect — in consultation with other key stakeholders of the system. The conventions by which a view is depicted will vary with the particular architectural technique or method employed; however, the engineering principles governing views may be generic — for example, completeness, the principle that an architectural view should represent the whole system from a single, well-defined perspective, is adopted in various methods [20, 25].

The notion of "components" recurs throughout systems and software thinking about architecture, as a construct for capturing the major elements of the "structure or construction" of a system. Of course, these elements do not exist in isolation but, as Meszaros observes, are related. The APG framework postulates two kinds of relations: connections (among components), and constraints (on components, or connections).

The Architecture Planning Group chose to define these terms as follows:

- *Components* are the major structural elements in a view; such as functions in a functional view, data models in a data view, or hardware in a physical view.
- *Connections* are the major relations between components of a view. They may be "run-time" relationships like control or data flow, or other dependencies.
- *Constraints* represent laws the system must observe; constraints apply to components and connections. There are three kinds of constraints:
  - Constraints reflecting performance, functional, or non-functional requirements (such as security, fault tolerance, quality)
  - Style and protocol rules, and
  - Laws of nature which constrain resources

The use of components and connections (or connect*ors*) seems to be a *de facto* standard in current architectural thinking, although the community is split on the latter term. Abd-Allah, Boehm, Gacek, Kazman, Abowd, Bass, for example, use "connection." Garlan, Shaw, and Allen use "connector." Luckham uses both terms to distinguish the relation between two components (a connection) from the entity which realizes that relation (the connector). In the present document, we use *connection* to encompass both notions, recognizing that we may want to refine this at a future time.

2.1 Uses of Architectural Description

To identify architectural concepts and principles relevant to the potential users of a future standard, we realized we needed to understand the roles which architectural descriptions may play within system life cycles. While development of a full "concept of operations" for architectural descriptions is left to the Working Group, the APG identified two fundamental ways of applying architectural precepts: *architecture as design* and *architecture as style*.

The first way is to use an architectural description as the vehicle for expressing high-level system characteristics that define and organize its major elements and their interrelationships. The architectural description is used to communicate between client and developer to aid clarification of requirements and assess their impact on system design. The architectural description is often developed through an evolutionary process from the initial expression of a system concept as a high-level abstraction to one of a more detailed and tangible expression that is widely accepted as being an expression of design.

The second way is to use a subset of the information used in a full architecture description to capture a style to facilitate certain common attributes among systems, ranging from system compatibility, interoperability, (component) interchangability, to (system) replacability. An *architectural style* is a set of patterns or rules for creating one or more architectures in a consistent fashion. There are many ways to capture and communicate a style [1, 2]. A *reference model* can be used to embody a style. Style is a partial characterization of a system; it does not represent the complete architecture for a system, but is a template for specifying the architecture of a specific system.

Architecture as design is useful for individual product development, analogous to the design of individual buildings, whereas *architecture as style* is also useful for harmony

among products, analogous to the design and planning of cities. Consider the following examples:

1. Individual software products have usually had an architectural concept established prior to implementation. However, all too often, conflicts between immediate user requirements and the architecture are resolved in favor of the requirements. The architecture is compromised over time, making the product less tolerant of modification or enhancement. More prominent attention to architecture can and would make software products more manageable over their life-cycle.

2. The architectural design of a new system can often benefit from an understanding of previous architectural designs for similar or related systems. However, the relative merits of one architecture vs. another for addressing a specific constraint varies according to the mix of other constraints upon the new system. A systematic approach to architectural description would aid this understanding by facilitating "reuse" of architectural knowledge.

3. Modern software development practices have evolved an even stronger impetus to adopt attention to architecture. In recent times, the state of the software practice has begun to include reuse of software products. Such reuse is only possible when expected behavior is consistent with actual behavior. As such, encouraging the software community to articulate and observe architectural style rules is likely to facilitate the further reuse and the continued maturation of software engineering as a discipline.

4. While it is important for an architect to understand the functional aspects of a system, it may be critical for the success of that system to embody an architectural view of those who must interface with it. If we look at a mass transit map of any major metropolitan area we see how various subsystems interconnect, where various stops are located, and perhaps travel times between locations. We would not, however, attempt to use the map for determining distance, or our precise location from a geometric perspective. Such maps are neither rendered to scale nor do they show every turn, rise, or fall. Yet, the view is critical for the successful use of the system.

5. In recent years Planned Unit Developments (PUD) have gained in popularity. From an architectural perspective the components that make up the unit are the various entities within the proposed development area. This allows the abstraction of many facets: roads (interfaces), electric, water (support services), population density (complexity).

6. Most major appliances have located on them a block diagram. For instance, a refrigerator might have a diagram on the back depicting the compressor, defroster, lights, temperature control, water location for ice maker, power cable, etc. Although not to be confused with the refrigerator itself, the diagram is one view of the device.

These examples meet the principles stated above, in that that they convey specific perspectives, dealing with domain concerns. In each case what has been described as an end product descriptor, probably had its beginning as a development document to convey conceptually what the device, product, or service was going to look like.

3. APG Recommendations

The primary result of our deliberations was an Action Plan, delivered to SESC. That Action Plan made several recommendations.

First, the Architecture Planning Group recommended that SESC approve two new Project Authorization Requests (PARs) for:

1. a *Recommended Practice for Architectural Description*; and

2. a *Guide* to the new Architectural Description standard.

Second, recognizing that architectural issues were of a wider scope than any single IEEE standard, the APG recommended that the APG Action Plan be disseminated to the SESC Working Group Chairs in order for the Chairs to review and apply the architecture precepts set forth therein for the generation and revision of IEEE Software Engineering Standards.

Third, the APG recommended that the APG Action Plan be disseminated to other affected disciplines so that the framework of terminology and architectural precepts set forth therein can be useful in establishing a dialog with others in those disciplines in order to coordinate their responses during the standards development process.

Fourth, the APG recommended that the Architecture Working Group established by the above PARs address the following topics in the preparation of the *Recommended Practice* and *Guide*:

- How does architecture fit into life cycle?
- How do architecture documents relate to other life cycle documents?
- How are architectures documented?
- Who are the stakeholders for an architecture?

- What architectural methods or processes are defined?
- What kinds of analyses may be applied to architecture models?

Finally, the Architecture Planning Group recommended that the following "concepts of operations" for an architect be explored and supported by the resulting Standard and Guide:

1. Software architects will require projections of available technologies to plan not a point solution, but a means to evolve a system including the user.

2. Architectural planning will include teams of experts in the relevant engineering disciplines of hardware, software, and human factors. For example, just as hardware should be selected which supports good software engineering, the software architecture should allow hardware evolution.

3. Time-to-complete a system will be balanced against time-to-obsolescence. System capabilities will be made operational and evolved iteratively through simulation if necessary. Too often, designs of the seventies are implemented in the nineties chasing a technology that will never be in use — obsolete before it is implemented.

4. Maturing the concept of prototype, architects will plan systems to include the user through early implementation of capabilities in the evolution to system solutions. Planning must continue throughout system evolution so that future capabilities are commensurate with the hardware engines available to support them. Like the weather, prognostication of the availability of storage, speed, and input/output devices will be necessary to judge whether the system will be reasonably current when implemented. A reasonable time for implementation can then be determined, and a reasonable subset of capabilities can be designated initial. Systems are not just architected as designs, they are planned to have initial capabilities which will evolve to solutions including user inputs.

5. Architects will plan with logistics, support, evolution, and continuing quality.

6. The expression of architecture will provide for the conveyance of lessons learned, suitability, etc., for the use of architectural solutions.

4. Conclusion

SESC has determined that there is sufficient technical basis and community interest in architecture to implement the Architecture Planning Group's recommendations for an Architecture *Working* Group which will undertake the definition of a *Recommended Practice for Architectural Description*, and companion *Guide*. The first two quarterly meetings of the Working Group took place in May and July 1996. The IEEE Architecture Working Group may be contacted at: `ieee-awg@spectre.mitre.org`.

Acknowledgments Philippe Kruchten (Rational) provided helpful comments on a draft of this paper as a member of the Architecture Working Group.

References

[1] A. Abd-Allah and B. W. Boehm. Reasoning about the composition of heterogeneous architectures. Technical Report USC–CSE–95–503, University of Southern California, 1995.

[2] G. Abowd, R. Allen, and D. Garlan. Using style to understand descriptions of software architectures. *ACM Software Engineering Notes*, 18(5):9–20, December 1993.

[3] B. W. Boehm. Software process architectures. In *Proceedings of the First International Workshop on Architectures for Software Systems*, Seattle, WA, 1995.

[4] Defense Information Systems Agency. *Technical Architecture Frameworkfor Information Management (TAFIM)*, version 2.0 edition, 1995.

[5] L. Druffel, N. E. Loy, R. A. Rosenberg, R. J. Sylvester, and R. A. Volz. Information architectures that enchance operational capability in peacetime and wartime. Technical report, US Air Force Science Advisory Board, February 1994.

[6] D. E. Emery, R. F. Hilliard II, and T. B. Rice. Experiences applying a practical architectural method. In A. Strohmeier, editor, *Reliable Software Technologies–Ada-Europe '96*, number 1088 in Lecture Notes in Computer Science. Springer, 1996.

[7] C. Gacek, A. Abd-Allah, B. K. Clark, and B. W. Boehm. On the definition of software system architecture. In *Proceedings of the First International Workshop on Architectures for Software Systems*, Seattle, WA, 1995.

[8] D. Garlan, editor. *Proceedings of the First International Workshop on Architecture for Software Systems*, Seattle, WA, April 24–25 1995. Published as CMU–CS–TR–95–151.

[9] D. Garlan. . Research directions in software architecture. *ACM Computing Surveys*, 27(2):257–261, 1995.

[10] *IEEE Std 1016, Recommended Practice for Software Design Descriptions*, 1987.

[11] *IEEE Std 610.12, Glossary of Software Engineering Terminology*, 1990.

[12] R. F. Hilliard II, T. B. Rice, and S. C. Schwarm. The architectural metaphor as a foundation for systems engineering. In *Proceedings of Sixth Annual International Symposium of the International Council on Systems Engineering*, 1996.

[13] I. Jacobson, K. Palmkvist, and S. Dyrhage. Systems of interconnected systems. *ROAD*, May–June 1996.

[14] P. B. Kruchten. The 4+1 view model of architecture. *IEEE Software*, 28(11):42–50, November 1995.

[15] H. W. Lawson, W. Rossak, and H. R. Simpson. Working group report – CBS architecture. In *Proceedings of the 1994 tutorial and workshop on systems engineering of computer-based systems*, Los Alamitos, CA, 1994. IEEE Computer Society Press.

[16] D. C. Luckham, J. Vera, and S. Meldal. Three concepts of system architecture. Technical Report CSL-TR-95-674, Stanford University, July 1995.

[17] M. W. Maier. Architecting principles for systems-of-systems. In *Proceedings of Sixth Annual International Symposium of the International Council on Systems Engineering*, 1996.

[18] M. W. Maier. System architecture: An emergent discipline? In *1996 IEEE Aerospace Applications Conference Proceedings*, volume 3, pages 231–246, 1996.

[19] G. Meszaros. Software architecture in BNR. In *Proceedings of the First International Workshop on Architectures for Software Systems*, 1995.

[20] M. Moriconi and X. Qian. Correctness and composition of software architectures. In *Proceedings of ACM SIGSOFT '94: Symposium on Foundations of Software Engineering*, New Orleans, LA, December 1994.

[21] D. Perry and A. Wolf. Foundations for the study of software architecture. *ACM SIGSOFT Sofware Engineering Notes*, 17(4), October 1992.

[22] E. Rechtin. *Systems architecting: creating and building complex systems*. Prentice Hall, 1991.

[23] D. T. Ross, J. B. Goodenough, and C. A. Irvine. Software engineering: process, principles, and goals. *IEEE Computer*, 8(5):17–27, May 1975.

[24] SofTech. Integrated Computer-Aided Manufacturing (ICAM) Task I – Final Report. Technical Report AFML–TR–78–148, Air Force Materials Laboratory, 1978.

[25] J. F. Sowa and J. A. Zachman. Extending and formalising the framework for information systems architecture. *IBM Systems Journal*, 31(3):590–616, 1992.

A. Architecture Planning Group Charter

The Architecture Planning Group will define for the Software Engineering Standards Committee the statement of direction for incorporating architecture into the set of IEEE standards for software engineering. Every system or subsystem has an architecture, as defined by IEEE 610.12–1990. Every system with software has a software view of that architecture. This planning group will define terms, principles and guidelines for software architecture, not in isolation, but integrated with the views of other disciplines.

Planned Tasks

1. Define a framework for relating the concept and principles of software architecture to software and systems engineering.

2. Examine selected IEEE software engineering standards for:

   a. their conformance to the framework of software architecture that has been defined in task 1.

   b. their applicability and use in Software architecture.

3. Produce an Action Plan with recommendations for incorporating software architecture into IEEE standards for software engineering. (E.g., obsolete old standards, modify existing standards, propose new standards). Provide recommendations for the Software Engineering Standards Committee to work effectively within the systems communities.